VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

TEMPERATURE MONITORING VIA ETHERNET

SLEDOVÁNÍ TEPLOTY PO SÍTI ETHERNET

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. RANNY KHAIT
AUTHOR

VEDOUCÍ PRÁCE — Ing. JIŘÍ SEKORA
SUPERVISOR

BRNO 2012

BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering and Communication

Department of Biomedical Engineering

Diploma thesis

Master's study field

Biomedical and Ecological Engineering

Student: Bc. Ranny Khait ***ID:*** 78262

Year of study: 2 ***Academic year:*** 2011/12

TITLE OF THESIS:

Temperature monitoring via Ethernet

INSTRUCTION:

1) Perform literature review and study the possibility of data transmission via Ethernet. 2) Choose an Ethernet module and a microcontroller suitable for realization of data transmission via Ethernet. 3) Design a device based on the Ethernet module and the microcontroller. Design a block diagram, electrical circuit and make a list of electronic components. 4) Choose suitable temperature sensors for measuring temperature in the range of -20° C to +60° C with an accuracy of 0.5° C. 5) Assemble a program for the Ethernet module and the microcontroller for temperature data transfer. 6) Implement the designed device with the power supply using PoE technology. Realize a data transfer test and evaluate the results.

REFERENCE:

[1] ENC28J60 Stand-Alone Ethernet Controller with SPI™ Interface [pdf]. Dostupný z <http://www.microchip.com/>.

[2] KABELKOVÁ, A., DOSTÁLEK, L. Velký průvodce protokoly TCP/IP a systémem DNS. Computer Press, 2008. 488 s.

Assigment deadline: 6.2.2012 ***Submission deadline:*** 18.5.2012

Head of thesis: Ing. Jiří Sekora

Consultant:

prof. Ing. Ivo Provazník, Ph.D.

Subject Council chairman

VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav biomedicínského inženýrství

Diplomová práce

magisterský navazující studijní obor
Biomedicínské a ekologické inženýrství

Student: Bc. Ranny Khait ***ID:*** 78262
Ročník: 2 ***Akademický rok:*** 2011/12

NÁZEV TÉMATU:

Sledování teploty po síti Ethernet

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte literární rešerši a prostudujte možnosti datového přenosu po síti Ethernet. 2) Vyberte vhodný modul a vhodný mikrokontrolér pro realizaci přenosu informací po síti Ethernet. 3) Proveďte systémový návrh zařízení postaveném na vybraném modulu a mikrokontroléru. Navrhněte blokové schéma zapojení, elektrické schéma zapojení a vytvořte rozpisku součástek. 4) Vyberte vhodné teplotní senzory pro měření teploty v rozsahu -20°C až +60°C s přesností alespoň 0,5°C. 5) Sestavte program pro modul a mikrokontrolér sloužící k přenosu teploty. 6) Realizujte navržené zařízení. Napájení zařízení bude pomocí technologie PoE. Realizujte testovací datový přenos a výsledky vyhodnoťte.

DOPORUČENÁ LITERATURA:

[1] ENC28J60 Stand-Alone Ethernet Controller with SPI™ Interface [pdf]. Dostupný z <http://www.microchip.com/>.
[2] KABELKOVÁ, A., DOSTÁLEK, L. Velký průvodce protokoly TCP/IP a systémem DNS. Computer Press, 2008. 488 s.

Termín zadání: 6.2.2012 ***Termín odevzdání:*** 18.5.2012

Vedoucí práce: Ing. Jiří Sekora
Konzultanti diplomové práce:

prof. Ing. Ivo Provazník, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

ABSTRACT

The main topic of the presented thesis is temperature monitoring via Ethernet. This thesis contains theoretical part, in which are described different types of sensors, temperature measurement, Ethernet, and system design and the selection of components. The practical part is focused on the implementation of the system, a description of the program and sending the measured values via Ethernet.

KEYWORDS

Ethernet, sensor, measurement, temperature, microcontroller

ABSTRAKT

Hlavním tématem práce je měření a přenos teploty po síti Ethernet. Práce obsahuje teoretický úvod, ve kterém jsou popsány různé typy senzorů, měření teploty, Ethernet a návrh systému a výběr součástek. Praktická část je zaměřená na realizaci systému, popis programu a odesílání změřených hodnot po síti Ethernet.

KLÍČOVÁ SLOVA

Ethernet, senzor, měření, teplota, mikrokontrolér

EXTENDED ABSTRACT

Diplomová práce se zabývá měřením a odesíláním teploty po síti Ethernet. Práce obsahuje teoretický úvod, ve kterém jsou popsány různé typy senzorů, měření teploty, Ethernet a realtime Ethernet a návrh systému. Obsahuje i popis všech vybraných částí k realizaci systému – mikrokontroléru, teplotního senzoru, Ethernetového řadiče a technologie PoE. Praktická část práce je zaměřená na popis celého programu sestaveného v jazyce C, dílčí části programu řešící inicializaci LCD a teplotních senzorů, uložení hodnot teploty, komunikaci s Ethernetovým modulem a odesílání změřených hodnot po síti Ethernet.

KHAIT, Ranny. *Sledování teploty po síti Ethernet:* diplomová práce. *Brno*: Vysoké učení technické v Brně. Fakulta elektroniky a komunikačních technologií. Ústav biomedicínského inženýrství, 2012. 51 s., 3 přílohy. Vedoucí práce Ing. Jiří Sekora.

DECLARATION

I hereby declare that I worked on this thesis independently, without outside help and that I used only literature listed in the Literature section. Like author this thesis I also hereby declare that I'm not creating this thesis infringe the copyrights of third parties, in particular, not infringe upon the rights of a foreign copyright personality in an illegal way and I am fully aware of the consequences of a breach of the provisions of Section 11 and the Copyright Act No. 121/2000 Coll., including possible criminal consequences resulting from the provisions of Section 152 of Criminal Law No. 140/1961 Coll.

Brno 18th May 2012 ....

author's signature

ACKNOWLEDGEMENTS

I would like to express my deep gratefulness to Ing. Jiří Sekora for supporting me in my work, for his patience and motivating me and valuable advices. I would also like thank to my family, especially my parents, for caring and for giving me the opportunity to study.

Brno 18th May 2012 ....

author’s signature

CONTENTS

INTRODUCTION

The aim of this diploma thesis is to design a device for measuring of temperature and transmission of obtained information over Ethernet controlled by a microcontroller and a powered through data cable using Power over Ethernet technology.

The introduction is dedicated to study of measuring of temperature and communication via Ethernet, following with selection of suitable types of sensors. The assignments requirement was to measure temperature in range of -20 ° C up to +60 ° C with precision of 0.5 ° C, which will be taken into consideration when choosing the appropriate sensor. The work also incorporates the selection of suitable controller for communication via Ethernet and directing microcontroller and design of manner of remote powering using Power over Ethernet (PoE) technology.

The practical part of diploma thesis contains design of system equipment and electric diagram of components and the actual compiling of the program. The controlling program for the microcontroller was compiled in C language and contains in addition to peripheral service (sensors and display devices) a simple web server with a page showing the values of temperature sensors.

1 TEMPRERATURE AND TEMPERATURE MEASUREMENT

This chapter discusses the temperature physical variables, the principle functions of temperature sensors and types of sensors. Thereafter sensors with readout in form of electrical signal are included in this section.

1.1 TEMPERATURE

Temperature is a fundamental physical quantity of the SI (International System of Units) which belongs to the thermodynamic industry. It is one of the principal thermodynamic properties because it occurs in many physical laws and it plays an important role in all fields of natural science including physics, geology, chemistry, atmospheric science and biology.

Temperature is directly proportional to the kinetic energy of substance molecules. Among the molecules leads to interaction and precipitation, therefore they change their speed. This speed is individual for each molecule and variable in time. If the substance is a solid state, the molecules will be oscillating chaotically around the equilibrium position. In case it is in a liquid phase, the oscillations will be irregular in the whole volume of liquid. If the matter isn't affected by surrounding environment, then the average speed of the molecules is constant and depends directly proportional to temperature. The lowest temperature is called *Absolute zero (0K = 273,15^{o}C)*, it's a condition in which there is no movement in matter.

Official and science unit of temperature is Kelvin (K), the less used one is Celsius (^{o}C). Conversion between these two scales is in the equation below by another unit for expressing the temperature ϑ[^{o}C], to this value can be approached, but it can't be exactly reached [1].

$$\vartheta = T - T_o,\ T_o = 273.15K \text{ [2]}$$

1.2 SENSOR TYPES

A sensor which is also known as *detector* or *converter*, that measures a physical, chemical or biological quantity and converts it into measurement quantity. The status of the monitored variable is captured by a sensitive part described as *receptor*, and sensor processing circuit evaluation. Output of information from the sensor circuit evaluation is quantitative, usually an electrical signal that can be handled by other control circuit or by an electronic instrument.

In general, there are many aspects under which sensors split into different categories and groups. The most common division is based on the idea that the sensor can be used as a converter between the stimulus expressed measured (detected) and output variable. Therefore, this idea comes with the following division:

1) By the input variables – sensors are divided by type of measured value. Here the measurement sensors are distinguished into:
   - Temperature values,
   - Geometrical quantities,
   - Mechanical quantities,
   - Electrical and magnetic quantities, etc.
2) By the conversion and transfer principle – sensors are divided according to the principle of transfer measured values to the variable output. Therefore, sensors are divided by:
   - Physical conversion,
   - Chemical conversion,
   - Biochemical conversion.
3) By the connection between the sensor and the measured environment:
   - Contact,
   - Contactless.
4) By the input power, into two categories:
   - Active, it doesn’t need external supply to be activated,
   - Passive, powered by an external supply
5) Otherwise, are specialized by the output, (analog - amplitude, frequency, digital).

Temperature sensors work based on several principles so the readout has different forms. It is indirect measurement because of the need to transform temperature to other value which we are able to evaluate. Only the sensors with readout in form of electrical signal are included in this section. The sensors can be divided according to their function into active which serve as power source (thermoelectric generator), and passive which must be powered from outside source.

1.3 THERMOELECTRIC GENERATORS

Thermoelectric generators (also called thermo-generators) demonstrated in figure (1) are devices which convert heat (temperature differences) directly into electrical energy, using a phenomenon called the Seebeck effect (or thermoelectric effect) which is generation of thermoelectric voltage between outputs composed of two different metals proportionally to difference between temperatures of junction point of two metals and reference point. If the temperature of junction point is invariable, the voltage of thermo-generator will be proportional to temperature of measuring end. The sensitivity of thermo-generator is in magnitude 10 μV/^{o}C. It can be enhanced by in serial connected several thermo-generators.

The advantage is great temperature range in which thermo-generators can function. This enables its wide application in industry for high temperature measurements. On the other hand, the precision in comparison with other systems is lower. Even thought it can reach ±0.5^{o}C for the most precise thermo-generators.

The thermo-generators can be used to generate electric current but their efficiency and output is low. To do so, it utilizes Peltier effect when the flow of current generates temperature difference between two ends of conductor.

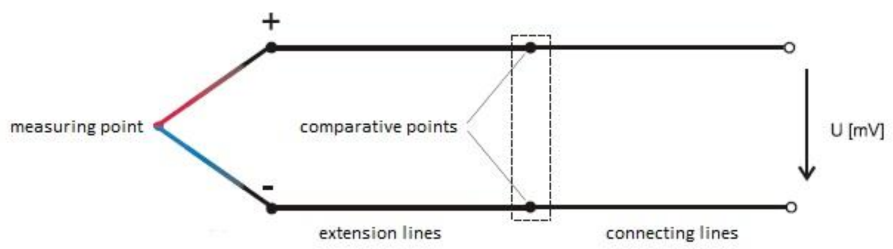


Figure (1): Thermoelectric generator [1]

1.4 METALIC RESISTANCE TEMPERATURE SENSOR

Resistance thermometers (RTh) are sensors used to measure temperature by correlating the resistance of the RTh metal with temperature. This correlation is defined by temperature coefficient of resistance α [K^{-1}]. The following linear equation (1.1) can be used with slight uncertainty for thermometers with range from 0^{o}C up to 100^{o}C.

$$R(t) = R0(1 + \alpha t) \tag{1.1}$$

α – coefficient of resistance [K^{-1}],

$R0$ – resistance at 0^{o}C [Ω],

t – temperature [^{o}C],

Coefficient of resistance is usually defined in technical practice as follows:

$$\alpha = \frac{R_{100} - R_0}{100 R_0} \tag{1.2}$$

$R100$ - resistance in 100^{o}C [Ω].

For greater range of temperatures must be used non linear equation (1.3) in dependence on used metal. The platinum, nickel, copper, molybdenum and alloys of NiFe are the most often used materials for production of metalic resistance thermometers. Very advantageous properties of these thermometers are very good long-term invariability of their parameters and wide range of working temperatures.

Platinum is the most often used material for production of temperature sensors. The most often used sensors are marked as *PT100* with intrinsic value of resistance of 100Ω in 0^{o}C. Also Pt sensors with other values of resistance, namely 50, 200, 500, 1000 and 2000Ω are manufactured. Pure platinum is very stable element with precisely described correlation of resistance with the temperature. For meteorological purposes the platinum used must have purity higher than 99.999% [1]. Small disadvantage of these sensors is relatively small change of resistance in dependence on temperature as shown in figure (2).

$$R(t) = R0[1 + At + Bt^2 + Ct^3(t - 100)] \qquad (1.3)$$

t – temperature [^{o}C],

$R(t)$ – resistance at 0^{o}C [Ω],

A – linear coefficient of resistance [K^{-1}],

B – quadratic coefficient of resistance [K^{-2}],

C – 3. power coefficient of resistance (C=0 for $t>0^{o}$C) [K^{-3}].

According to the IEC treaty, the platinum resistance thermometers are divided into two groups in dependence to their tolerance to error. Group A is for range of temperatures from –200^{o}C up to 650^{o}C while group B is defined for ranges from –200^{o}C up to 850^{o}C. The error in 0^{o}C for group A is ±0.125^{o}C, which is correspondingly ±0.06 Ω, whereas for group B it is ±0.25^{o}C and ±0.12 Ω.

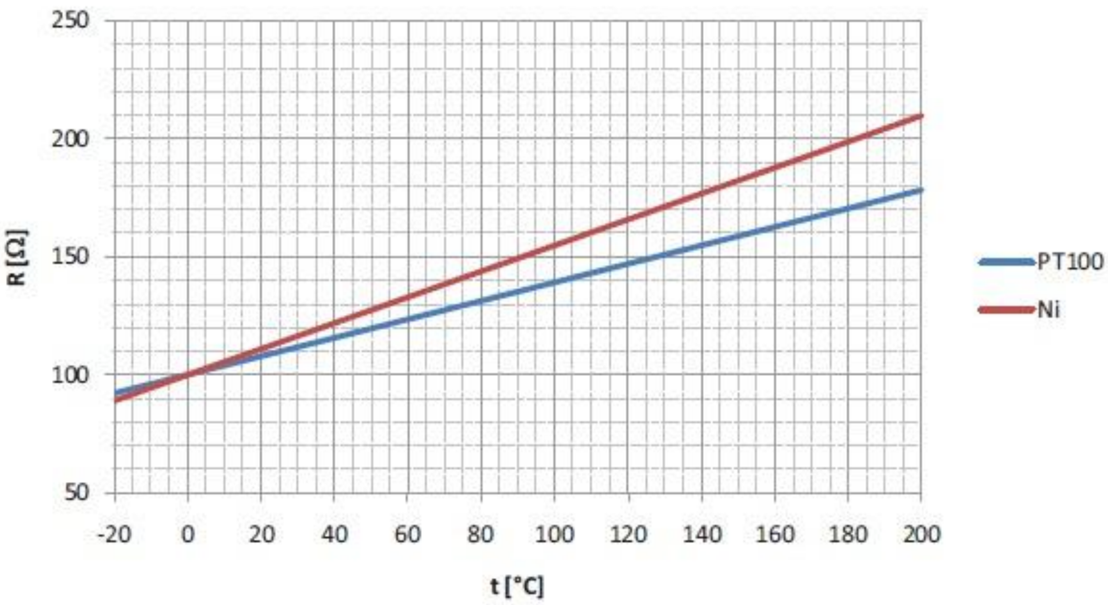


Figure (2): Resistance to temperature for nickel and platinum sensors [2]

The resistance temperature sensors are manufactured with two or four outputs. The resistance of the two outputs (of the two output sensor) is added to measured resistance which is the cause of variable error. This variability of resistance is in range from 0.1°C up to 0.5°C.

The basic of resistance temperature sensors is that they are heated by current flowing through them. The reader then indicates higher temperature than is the real value. On the other hand the low value of current induces only small change of voltage on resistance, which must be compensated by amplifying; this however boosts the error of sensor. The amplification of current must be resolved in such a way that takes into account both phenomena. The manufacturers of resistance temperature sensors denote the loading invariable D, which corresponds with value of wattage according to equation (1.4), which induces change of sensor's temperature about 1°C. The value of amplification current can be calculated according to the equation (1.5).

$$P = R(t)I^2 \tag{1.4}$$

P – power dissipation[W],

I – excitation current [A],

R(t) – sensor’s resistance [Ω],

$$I = \sqrt{\frac{D}{R(t)}} \tag{1.5}$$

D – loading constant [$W \cdot K^{-1}$],

1.5 SEMICONDUCTOR TEMPERATURE SENSORS

These sensors also utilize dependence of resistance on temperature. The concentration of electrons depending on temperature plays dominant role in semiconductors. Temperature coefficient of resistance of semiconductors is negative and has higher value than that of metals. Thermistor is the most basic semiconductor sensor. They are based on change of resistance in dependence on temperature of ceramic semiconductors. The advantages of these sensors are high temperature change of resistance and small proportions. The disadvantages are high non-linearity and worse invariability of parameters in time which is further decreased by artificial aging of sensor.

Thermistors are divided according to their structure into amorphous and polycrystalline. In dependence on structure, the thermistor has either high negative temperature coefficient of resistance – so called negastor (NTC thermistor) or high positive one making it so called pozistor (PTC thermistor).

1.5.1 NEGASTOR (NTC THERMISTOR)

Progress of resistance dependence on temperature in NTC thermistors is described by equation (1.6) [3]. It is polynomial approximation of progress of third magnitude. The coefficients are denoted by manufacturer for every type of thermistor. The working range of temperatures is from –80°C up to 120°C. Lesser working range of temperatures and durability are the main disadvantages of group of sensors. For industrial temperature measurement are utilized aforementioned techniques (thermo-generators, PT100). The precision varies according to used sensor but in general the thermistors are the most precise, for example thermistors from U.S. sensors have error ±0.05°C.

$$T = [A + Bln(R_T) + C(\ln(RT))^3]^{-1} \tag{1.6}$$

T – thermodynamic temperature [K],

A, B, C – constants provided by the manufacturer,

R_T – thermistor's resistance [Ω],

The convert of temperature to resistance is given by the following equation:

$$R_T = e^{\sqrt[3]{\beta - \frac{\alpha}{2}}} - e^{\sqrt[3]{\beta + \frac{\alpha}{2}}} \tag{1.7}$$

Where constants α and β are determined as follows:

$$\alpha = \frac{A - \frac{1}{T}}{C} \qquad ; \qquad \beta = \sqrt{\left(\frac{B}{3C}\right)^3 + \frac{\alpha^2}{4}}$$

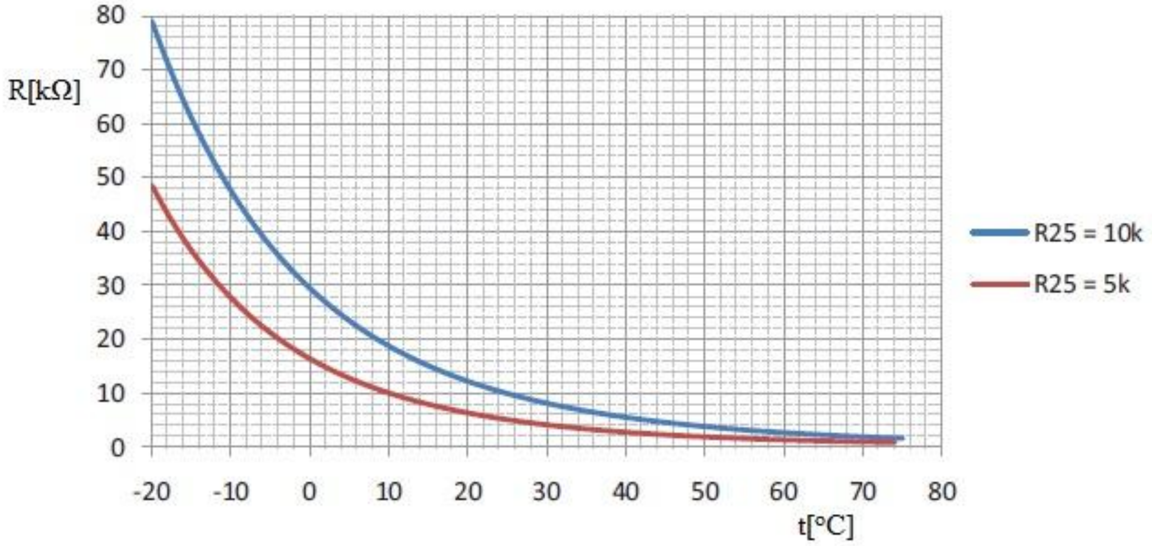


Figure (3): The relation between thermistor resistance and temperature

Progress of resistance dependence on temperature can be calculated according to equation (1.7) as shown in figure (3). The progress of resistance dependence on temperature of thermistors with intrinsic values 10kΩ and 5kΩ in temperature of 25°C are shown for illustration. The intrinsic value of thermistor is chosen according to the range of temperatures which are set as working parameters and to induce high enough change of voltage by current flowing through thermistor. This is the reason why are the thermistors with high value used in high temperatures and vice versa. The range of manufactured resistances is from 100Ω up to 100kΩ.

The thermistors are used in different configuration in dependence on purpose of usage. The thermistors provided with epoxide layer teflon tube are very common. The example of thermistor design is shown in figure (4).

The measuring bridge or four-wire connection with current source are used to evaluate the change of resistance. In this case, low current must be ensured for the thermistor, which can reach value around 10µA and is denoted by manufacturer or through loading invariable. The final value of current can be calculated according to the equation (1.5).

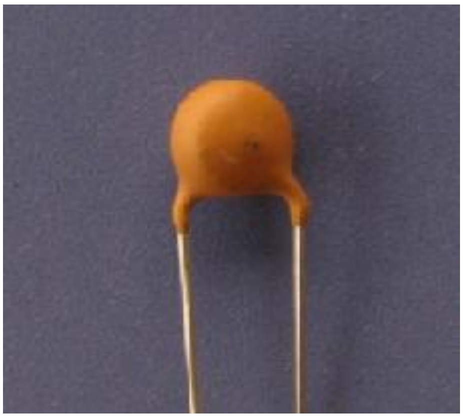


Figure (4): Thermistor design

1.5.2 POZISTOR (PTC THERMISTOR)

The pozistors are thermistors with high positive temperature coefficient of resistance made from polycrystalline ferroelectric ceramics, for example $BaTiO_3$. The resistance of pozistor is slightly lowered by increase of temperature until it hits specific value (called Curier's temperature) where the resistance is rapidly heighten in order of several magnitudes after which the value of resistance is again slightly lowered [2].

Pozistors are suitable for signaling of overreaching of allowed temperature of for example electric motors, where they are integrated into the winding. PTC thermistor can be used to limit current by circuit, where higher value of current flowing through circuit induces heating of the said circuit which therefore has higher resistance [4]. The pozistors can be used to measure temperature only in limited range of temperatures and the results should be regarded as unreliable because the progress of resistance dependence on temperature is not most of time precisely described by equation but graphically and therefore it is not possible to convert the resistance to temperature precisely.

1.5.3 SEMICONDUCTOR MONOCRYSTALLINE TEMPERATURE SENSORS

Monocrystalline temperature sensors are made from elements silicon, germanium or indium and their compounds. They enable detecting temperatures in range from -50°C up to 150°C (true for sensor KTY10). These sensors have lower precision but it is sufficient for manufacturing thermometers which have less demanding requirements. These thermometers are designed with encased sensor, for example TO-92. Their resistance dependence on temperature is described by equation (1.8) and also shown in figure (5).

$$R_{(t)} = R_{25}(1 + \alpha\Delta t + \beta\Delta t^2) \tag{1.8}$$

Δt – t – 25°C,

R_{25} – resistance at t=25°C [Ω],

α, β – coefficients of polynomial function.

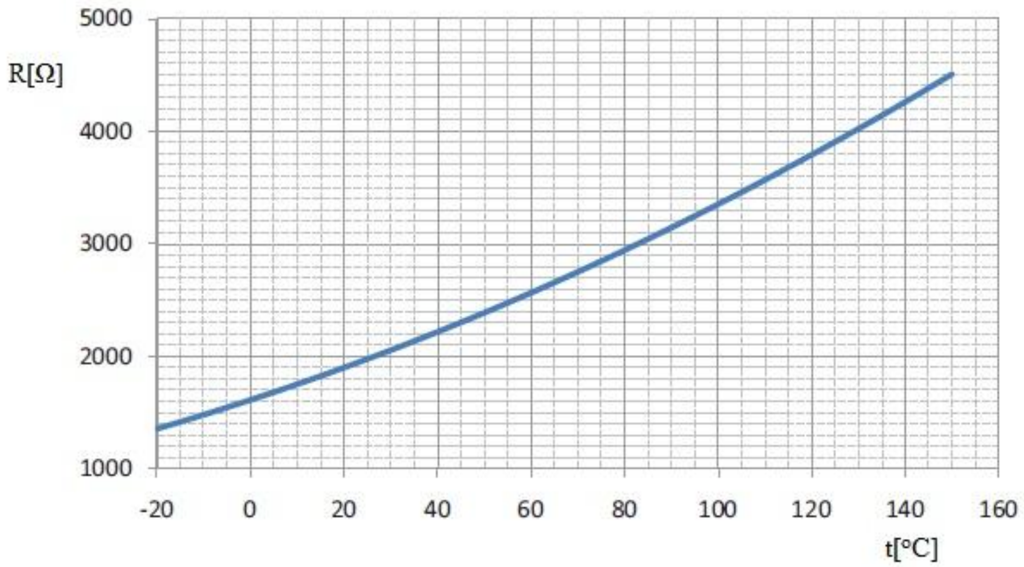


Figure (5): The relation between resistance and temperature of the silicon sensor KTY10-5

1.6 MONOLITHIC PN TEMPERATURE SENSORS

Semiconductor PN temperature sensors are based on dependence of voltage of PN junction in forward direction on temperature. The diode and transistor are type of PN junction device which utilize base-emitter junction. AV characteristic of junction for two different temperatures is shown in figure (6).

The particular change of voltage correlates to change of temperature in particular value of current. This phenomenon can be described by equations (1.9) and (1.10). This way of measuring temperature utilizes parasitic features. This enables to manufacture very cheap sensors, for example sensors for switching after reaching ceiling temperature. This phenomenon is also utilized in integrated sensors [1].

$$I_D = I_S(e^{\frac{U_D}{mU_T}} - 1) \quad (1.9)$$

U_D –voltage of PN junction in the forward direction,

I_S –saturation current of PN junction in the reverse direction,

I_D –saturation current of PN junction in the forward direction,

U_T –thermal voltage [J·C^{-1}],

m – recombination coefficient of semiconductors $(1 \leq m \leq 2)$.

For the voltage of PN junction is valid the following equation:

$$U_D = mU_T \ln\left(\frac{I_D}{I_S} + 1\right) \tag{1.10}$$

Thermal voltage is based on dependence temperature as follows:

$$U_T = \frac{kT}{e} \tag{1.11}$$

k – boltzmann constant $k = 1.38 \cdot 10^{-23}$ [J·K^{-1}],

T – thermodynamic temperature[K],

e – elementary charge $e = 1.602 \cdot 10^{-19}$ [C].

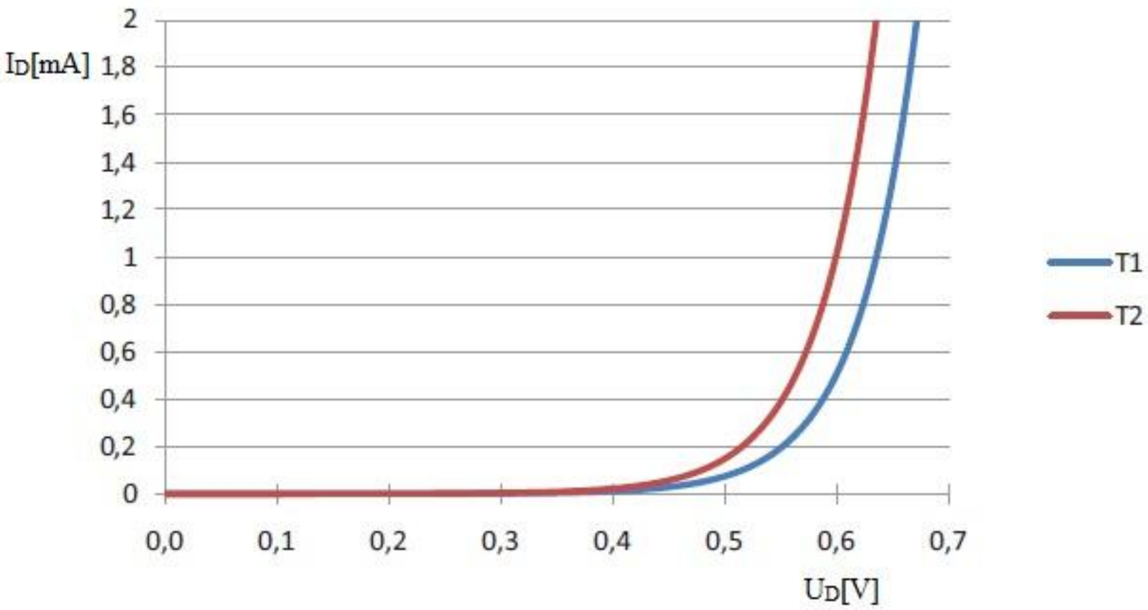


Figure (6): Temperature effect on AV characteristic of the PN junction T1<T2

2 NETWORK PROTOCOL

The Internet works on a hierarchical protocol store, as shown in table 1. The IP (Internet Protocol) layer is the conjunct layer to all Internet applications. This layer grants a connectionless, unreliable packet based delivery service. It was characterized as connectionless because packets are manipulated apart of all others. Because of no guarantee of delivery it is unreliable. Packets could be silently dropped, delayed, duplicated or can arrive squandered. This service is also called a best effort service. All attempts to hand over a packet will be made, with unreliability only caused by hardware mistakes or drained resources.

There are no simple methods to ensure a quality of service (QoS), as there is no purpose of a connection at the IP level. QoS is a request from an application to the network to ensure a guarantee on the quality of a connection. This lets out an application to appeal a fixed value of bandwidth from the network, and assume it is going to be executed, as soon as the QoS request has been accepted. Further a fixed delay, that is to say no jitter and in order delivery can be assumed.

A network that fuels the QoS will be secured from accumulation problems, in view of the network will shuffle of connections that request greater resources than can be supplied. Here is an example of a network which supports QoS, which is the casual telephone network, where each call is guaranteed the bandwidth for the call. Most users somewhere have heard the overloaded signal where the network cannot ensure the desired resource needed to make a call.

The application is the decision maker for which transport protocol is used. Both protocols shown here, TCP and UDP are the most usually used ones.

Table (1): simplified IP protocol stack [5]

<table>
<tr><td>Application</td><td>WWW</td><td>FTP</td><td>E-mail</td><td>NFS</td><td>VoIP</td><td>DNS</td></tr>
<tr><td>Transport</td><td colspan="3">TCP</td><td colspan="3">UDP</td></tr>
<tr><td>Network</td><td colspan="6">IP</td></tr>
<tr><td>Physical</td><td colspan="2">Ethernet</td><td colspan="2">AAL-5</td><td colspan="2">HDLC</td></tr>
</table>

2.1 REAL TIME ETHERNET

2.1.1 ETHERNET TODAY

An important progress of the technological evolution, are the real-time electronic distributed control system, which are used up to control and monitor precautionary applications such as flight board to hospital operating rooms. Seeing that real-time systems become more dominant and developed, as well as the requirement to physically extend the control in strict real-time. In such a way there is a need for control network protocols to promote strict real-time demands. These networks should ensure a guarantee of service in order to consonantly operate deterministically and rightly.

As defined in IEEE 802.3, Ethernet is un-necessarian, and then is inadvisable for hard real-time applications. The media access control protocol, CSMA/CD by its stand back algorithm, prohibits the network from boosting hard real-time communication due to its random retardations and potential transmission defection.

Growing demand and decreasing costs for separate network type, from boardroom to plant-floor, have conduced to the advancement of Industrial Ethernet. The aspire to incarnate a real-time element into this increasingly popular single-network solution has led to the advancement of different real-time Industrial Ethernet strategies. Process field bus networking standards have misfired to give over an integrated solution. Looming real-time Industrial Ethernet solutions supplement the fieldbus standards such as, by the use of casual user layers.

2.1.2 REAL-TIME INTODUCTION

According to figure (7), real-time (RT) systems are becoming more and more important, as industries concentrate on distributed computing in automation. As declining computing costs, and computing power escalates, industry has depend more on distributed computers to provide performance and increased decree to production lines. RT is not faster execution, but it is a process depending on progression of time for useful operation.

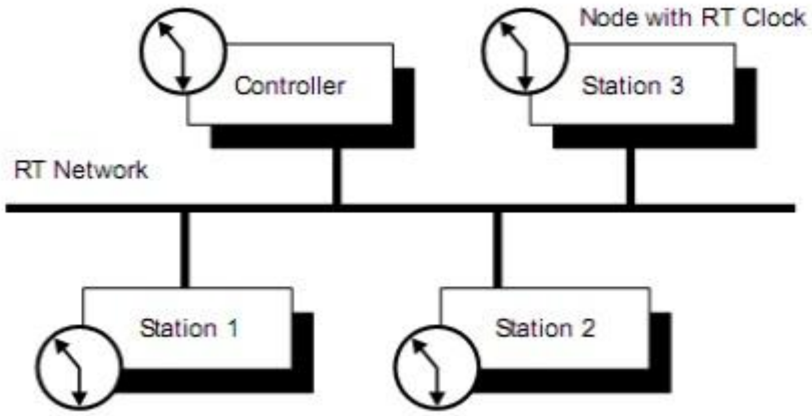


Figure (7): Distributed Real-Time processing [6]

RT systems are systems which depend not purely on the acceptance of data but also on its opportune. Correct RT system, is avouching successful system operation, as long as its timely execution is ruffled. In general, RT systems are separated into two divisions: hard and soft.

Hard Real-Time (HRT) are systems in which mistaken operation can get into tragically events. Fault in these systems can cause accidents or even death, for example train or flight control accidents.

Soft Real-Time (SRT) are systems which are not fragile. When not encouraged, a mistake will not bring on loss of life or fortune. Safety of SRT systems is not as critical as HRT systems, and should not be used in a safety critical situation. SRT systems examples are online reservation systems. RT system's building blocks are the **Jobs**. Every RT job has definite temporal quantities:

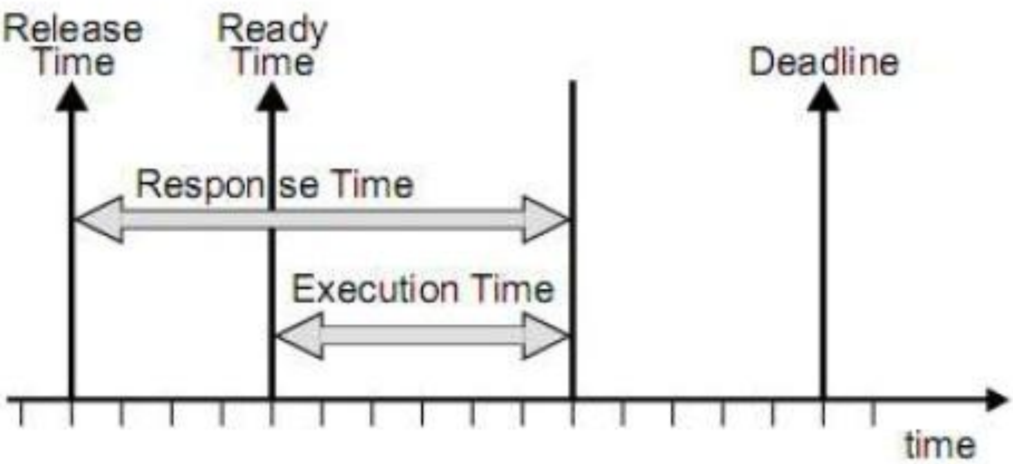


Figure (8): Temporal quantities of Real-Time job [6]

1. Release Time is, when a job becomes available to the system.

2. Ready Time is the earliest time a job can start executing.

3. Execution Time is the time for a job to be completely processed.

4. Response Time is the interval between release time and completion of the execution.

5. Deadline is the time by which execution must be finished, beyond which the job is late. A deadline can be hard or soft, recommending the job's temporal subservience. As referenced above, a waste hard deadline can have serious implications.

To generate a RT distributed system of network computers, it is fundamental to yield communication between the particular computers in a responsible and suitable way. Laid out processors leading RT applications should be able to connect via RT protocol, alternatively the temporal grade of work is missed out.

Real-Time communication networks are the same as any RT system. They also can be hard or soft, this depends on demands and their jobs involve message transmission, promotion, and reception. Quantities of RT control networks are used in industry, but neither have the favours or bandwidth of Ethernet.

2.1.3 THE REQUEST FOR REAL-TIME ETHERNET

Request for Ethernet as a RT control network is growing as makers understand the goods of using a single network technology as been said. Lowered product costs with potentiality of overlaying training and up-keeping costs for information, operation level, control and perhaps device networks would extremely decrease manufacturer's expense.

Pending RT control level, Ethernet suggests lots of initiatives over existing solutions. Like a control network, 10Gbps Ethernet gives bandwidth nearly 1000x more quick than today's reconcilable field-bus networks and can also under-prop RT communication. Distributed applications in control surroundings needs tight synchronization in order to undertake the delivery of control messages beyond established message cycle times (table 1). Field-bus systems and conventional Ethernet cannot comply cycle time needs below a few milliseconds, but looming RT Industrial Ethernet treatment provides cycle times of a few microseconds.

In conjunction with enhanced bandwidth and tight synchronization, RT Ethernet facilitate manufacturers the reliance of using a physical and data-link layer technology that has been normed by both the IEEE and the ISO. Ethernet can safeguard minimization with all the feature necessary of a field, control or tool network. Moreover, Ethernet devices can also under-set TCP/IP stacks in order that Ethernet can handily get in to the Internet. This release is enticing to employers hence it allows remote diagnostics, control and sighting of their operation network from any Internet-connected appliance around the world with a free license of web browser. For all Ethernet presents uppermost by its minimum message data size (46 bytes), that is great in comparison to given control network standards, its enhanced bandwidth, standardization and mergence with present technology should establish genial reasons to interpret Ethernet as a control network solution.

2.2 TRANSMISSION PROTOCOL TYPES

2.2.1 TRANSMISSION CONTROL PROTOCOL (TCP)

One of the main protocols in TCP/IP networks is the TCP. While the IP proceeds with packets only, TCP provides two hosts to start a connection and exchange streams of data. TCP warranty consignment of data, and likewise that the packets are going to be delivered as well as were sent and in the same order.

TCP is a connection-directed protocol which is accountable for dependable communication between two end processes. The element of data transmitted is called a stream, that is just a succession of bytes.

The meaning of being connection-directed or oriented is that ahead of starting transferring data, there is a need to open the connection between the two end points. Data transfer can be done in full duplex, which is sending and receiving on a single connection. After the transfer is done, the connection has to be close to free system resources. Both of the ends know when the transmission is started and when was ended. Before the agreement of both ends with the connection, the data transfer cannot come-true. The connection can be finished by whichever side, and the other is reported. This is made to end courtly or just abort the connection.

Entity stream oriented entails that the data is an annominate sequence of bytes. Nothing can make data dividing line obvious. The receiver has no case cognizant of how the data was transmitted indeed. Data can be sent as a big part and the receiver receiving it in some few smaller parts, or the sender can send many small data parts and the receiver receives only one big part. It can be guaranteed that all data sent are going to be received, empty of any error and in the right order. If any error happens, it will be corrected as a self-acting, in some cases transmitted again, when cannot be corrected it will be notified.

The TCP stream seems like a flat file, at the program level. While data is written to a flat file, and read back after, it's impossible to recognize if the data has been written in more small portions or in only one part. Until there will be added something special in order to recognize record boundaries.

At the programming level, it is completely simple to use TWSocket. For sending data there is a need to call the Send method in order to afford data to be transmitted. TWSocket will put it in a buffer unless it can be actually transmitted. In some case the data will be sent in the background and the OnDataSent event will be created when the buffer is emptied.

A program has to wait while it receives the OnDataAvailable event, to be able to receive data. On each occasion a data packet comes from the lower level this event is launching. The application has to call the Receive method to get the data from the low-level buffers. It's important to receive all the data available, to shirk from a never ending loop because TWSocket will start the OnDataAvailable again if all the data is not received.

As mentioned above, the data is a stream of bytes. Then, application should be ready to receive data as sent from the sender, divided into smaller parts or combined in bigger parts. For example, while sending two separated words, it is possible to get only one OnDataAvailable event and receive it as one word in one chunk, or to get two events, one for each word. The third possibility is to receive smaller chunks by fragmenting the words into smaller parts.

2.2.2 USER DATA PROTOCOL (UDP)

The UDP is a connectionless protocol which runs on top of IP networks as TCP. But unlike TCP/IP, it grants very few error reparation services, bidding instead a straight way to send and receive datagram across an IP network. It's used foremost for broadcasting messages over a network.

UDP makes out a connectionless host to host communication path. It has minimal overhead; every packet on the network is built up of a small header and user data, which is called a UDP datagram. Datagram boundaries between both the sender and the receiver are preserved in UDP. It entails that the receiver socket is going to receive an OnDataAvailable event for every datagram sent, and the Receive method will return a complete datagram for each call. In the case of too small buffer, the datagram will be abbreviated. When the buffer is too large, the left over buffer space is not touched and only one datagram is returned.

The UDP is connectionless, which means that a datagram can be sent at any time with no earlier sign, negotiation or adjustment. The datagram is just sent hoping the receiver is able to handle it. This protocol is untrustworthy because, it does not guarantee that the datagram is going to be delivered to the intended host. But it's important to know that the fault liability is very low on the Internet and almost null on a LAN excerpt that the bandwidth is full.

The datagram can be delivered in an incorrect order or undelivered. It means a possibility of receiving a packet before another one, even if the second was sent before the first. In some cases, the same packet can be received twice [5].

The main disadvantage for UDP is unreliability; therefore it is complicated to program it at the application level.

The main advantages are that datagram boundaries are recognized, the possibility to broadcast and that it's fast.

2.2.3 ADDRESSING

The same addressing scheme is used for both TCP and UDP. An IP address (32 bits number, always written as four 8-bit number couched in as unsigned 3-digit decimal numbers detached by dots) and a port number (16-bit number couched in as unsigned decimal number).

The low level protocol uses the IP address to route the datagram to the selected host on the determined network. After using the port number to route the datagram to the correct host process.

For both given protocols, a single host process exists at a time to receive data sent to the specified port. Commonly one port is dedicated to one process.

3 THE DESIGN OF THE SYSTEM

3.1 DESCRIPTION OF THE SYSTEM

In this part of work is illustrated the block diagram for temperature measuring system, figure (9). The pattern consists of partial blocks:

- Microcontroller
- Temperature sensors
- Display
- LEDs
- Ethernet controller with SPI interface
- PC
- Power supply
- PoE system

The heart of this measuring system is the microcontroller which will manage the entire system. Selection of a microcontroller should consider application requirements. It is necessary to include AD converters, in case of the use of sensors with analog outputs. Sensors which were chosen have digital outputs. LEDs and a display are implanted to control the work of the system and its improvements. It is also necessary to be able to increment the internal counter register with external pulse. For sending data vie Ethernet is and communication with PC is used Ethernet controller with SPI interface, which is powered by PoE system.

Communication with PC for monitoring measured values is realized by sending data to internet, by the used Ethernet controller and after setting IP address, will be displayed on PC. Data are sent synchronously after the measuring cycle. The communication protocol will use eight bit data word, which will include temperature values. If the data frame received by the PC are not attempting flawlessly, it will be ignored.

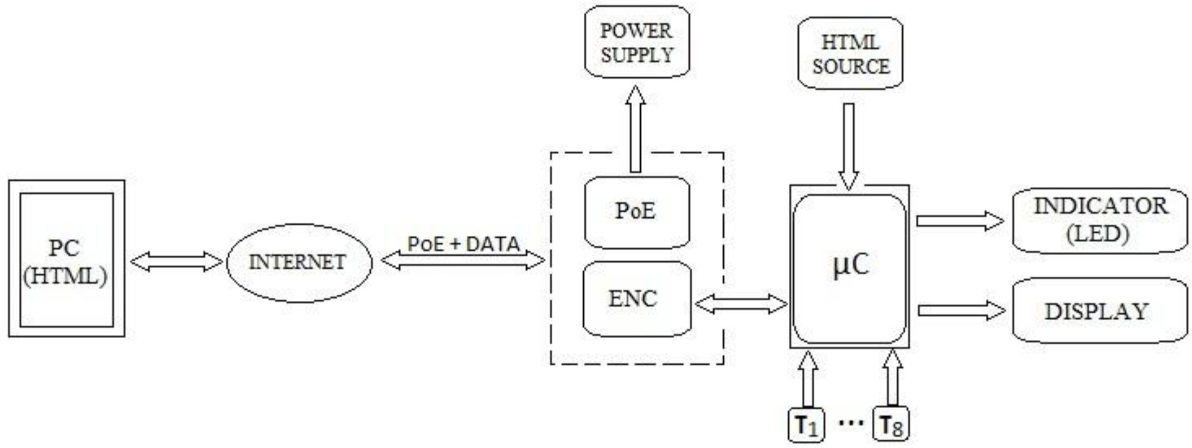


Figure (9): Block circuit diagram

3.2 SELECTION AND DESCRIPTION OF THE MICROCONTROLLER

Microcontroller selected for this work is from ATMEL; model ATmega644 as shown in figure (10), deployed on a development kit AND-Tech EvB 4.3 rev. 4. This is an encapsulation DIP40. Usually operates at frequencies up to 20 MHz. The development board was equipped with 16 MHz crystal, which was tested for timing delays and left on the final product. It contains 4 I/O ports, some of which have special functions. Some of microcontroller’s features are:

- 32 programmable I/O lines
- Operating voltage 2.7V up to 5.5V
- 64k bytes of In-system self-programmable Flash program memory
- 4k bytes EEPROM
- 8-channel, 10-bits analog to digital convertor
- Tow programmable serial USART
- Master/slave SPI serial interface

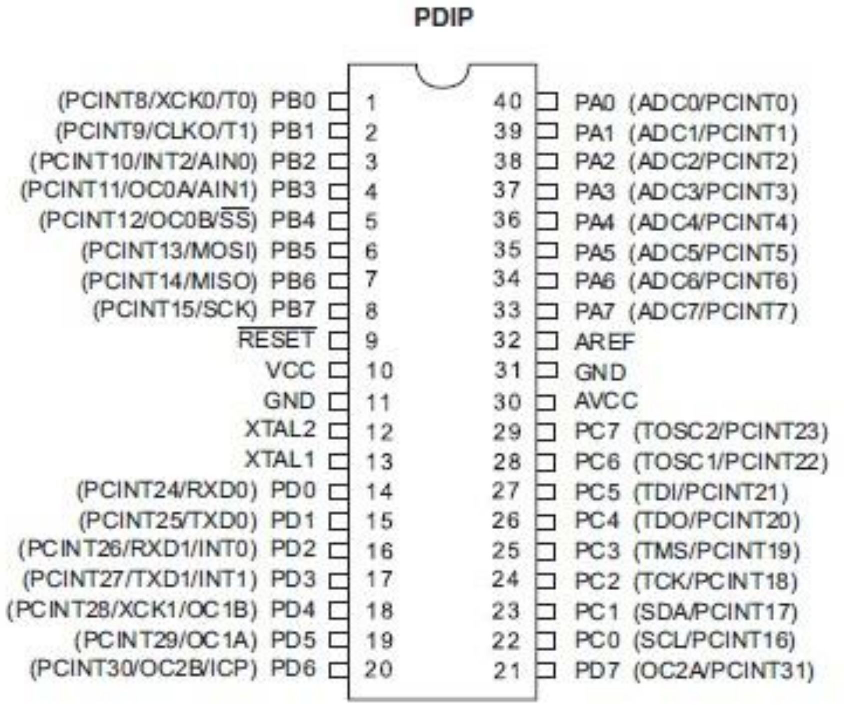


Figure (10): ATmega644 [8]

3.2.1 THE CORE OF AVR

Processor ATmega644 is based on the AVR core type RISC (*Reduces Instruction Set Computing*) Harvard architecture. It provides for the implementation of the program, access to memory, peripherals and service interruptions. Execution of instructions in the program is able with a single level pipelining. When one instruction is being executed, the next one is pre-fetched from the program memory.

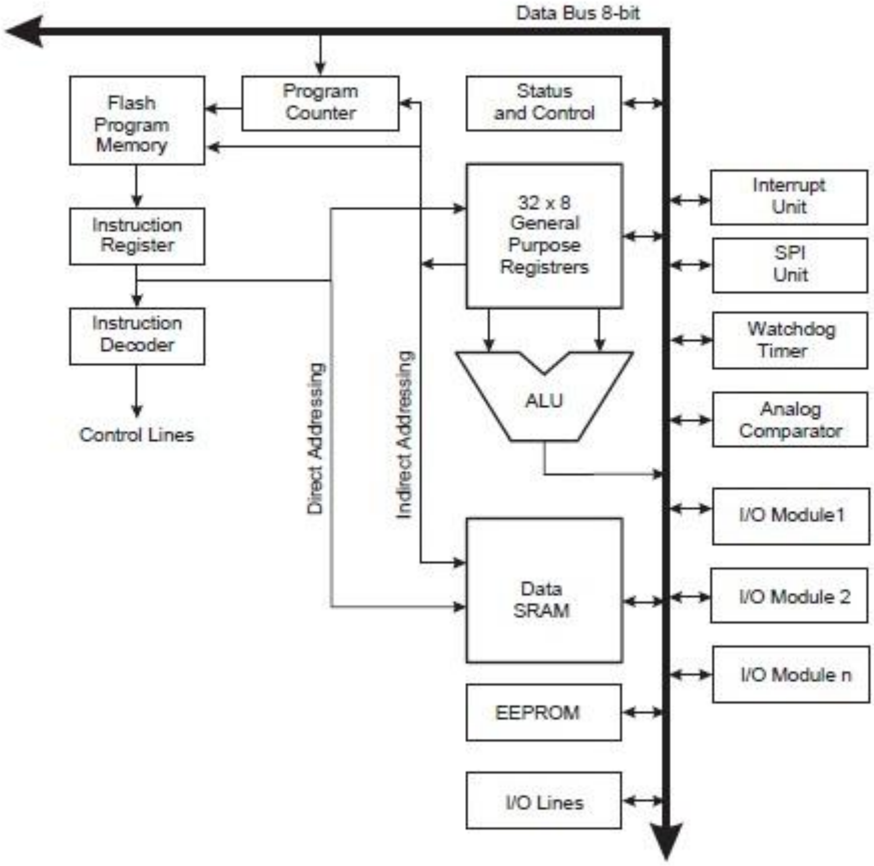


Figure (11): Block diagram of the AVR architecture [8]

AVR core consists of 32 identical 8-bit registers that can contain either data or addresses. Last 6 of the 32 registers can be used in pair as indicators of addresses for indirect addressing of data memory. These added function registers are the 16-bit X-, Y-, and Z- register. One of the three address pointers can also be used as an address pointer for look up tables in Flash program memory.

The programmer has the choice if the address pointer will be a processing instruction to increment or decrease prior to processing the instruction. It is useful for addressing the possibility to use 6-bit shift in the indicator’s address in the pair registers Y- and Z-.

AVR architecture has 5 modes for addressing data memory:

- Direct addressing,
- Indirect addressing ,
- Indirect addressing with 6-bit shift,
- Indirect addressing with decrement of pointer address, before processing instructions,
- Indirect addressing with address pointer incremented after processing instructions.

The register file is optimized for the AVR enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following I/O schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input,
- Two 8-bit output operand and one 8-bit result input,
- Two 8-bit output operand and one 16-bit result input,
- One 16-bit output operand and one 16-bit result input.

In figure (AVR CPU registers) is demonstrated the structure of the 32 general purpose working registers in the CPU (*Central Processing Unit*). Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions. Each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space.

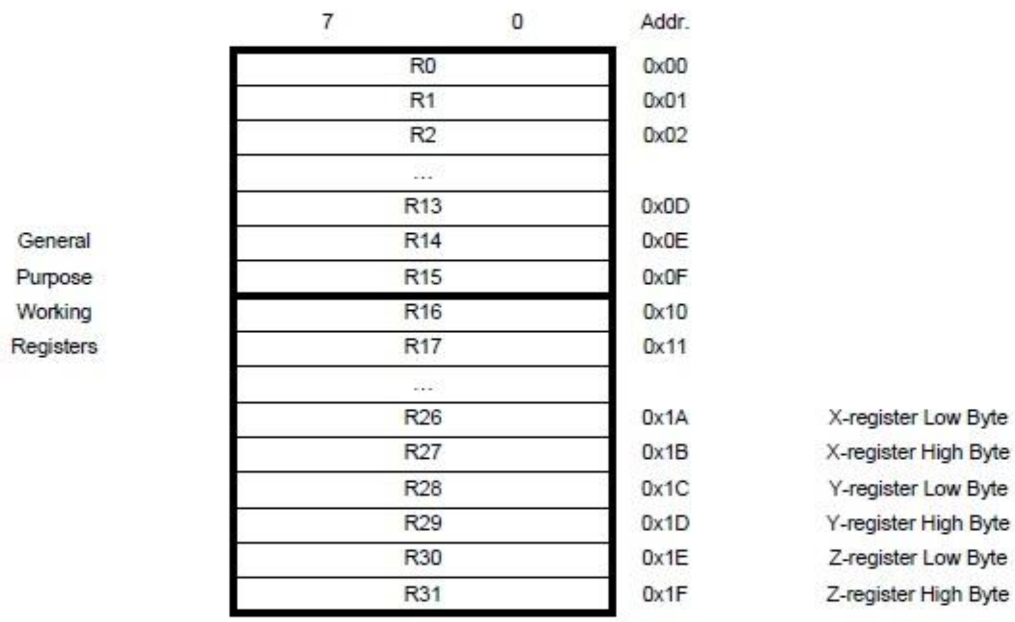


Figure (12): AVR CPU registers [8]

3.3 DIGITAL THERMOMETER

For temperature measurement was chosen digital temperature sensor DS18B20 from MAXIM Company, with a TO-92 case as shown in figure (13), and its block diagram in figure (14). This sensor was chosen for better and easier tasting, range of use, simple operation and according to its accuracy of results. Few selected advantages of the given sensor are:

- Unique 1-Wire interface, which needs only one Port Pin to communicate.
- Unique 64-bit serial code stored in an on-board ROM

- It needs no external components
- Power supply range from 3.0V to 5.5V, can be powered from data line
- Temperature sensing from -55^{O}C to +125^{O}C
- ±0.5^{O}C accuracy in a range from -10^{O}C to +80^{O}C
- Converts temperature to 12bit digital word in 750ms
- User selectable from 9 to 12 Bits for thermometer resolution

Figure (13): Digital temperature sensor DS18B20

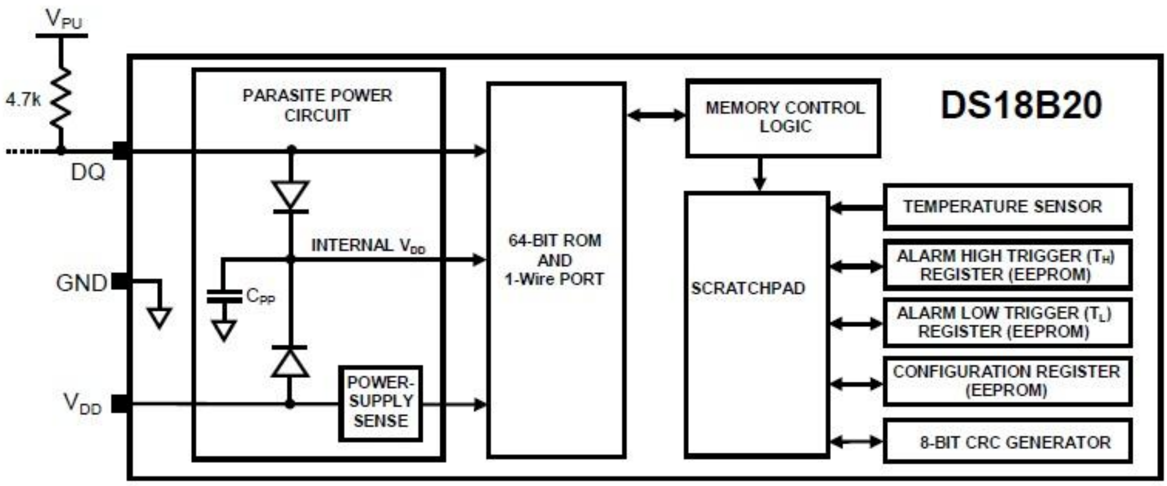


Figure (14): Block diagram of DS18B20 [9]

The digital thermometer DS18B20 has 3 wires (pins) described in table (2). To work properly there is a need to insert pull-up resistor of 4.7kΩ between the 5V power and data output, as shown in figure (15). This explains the use of 1-wire bus, where is a need of one wire only for communication.

Table 2: PIN description of TO-92

PIN	NAME	FUNCTION
1	GND	Ground
2	DQ	Data inout/output
3	V_{DD}	Optional V_{DD}, must be grounded for operation in parasire power mode

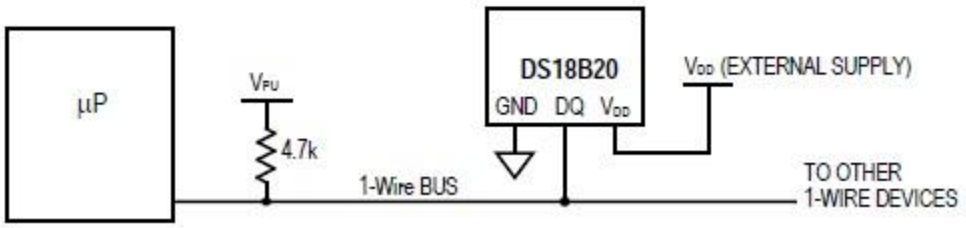


Figure (15): Powerinf the DS18B20 [9]

For safe and correct start of communication and temperature sensing, the DS18B20 should be reset by getting reset pulse from the master, where the bus master transmits (Tx) the reset pulse by pulling the 1-wire bus low for a minimum of 480µs. The bus master then releases the bus and change into receive mode (Rx). When the bus is released, the 4.7kΩ pull-up resistor pulls the 1-wire bus high. The DS18B20 waits 15µs to 60µs when it finds out this rising edge, then sends the presence pulse by pulling the 1-wire bus low for 60µs to 240µs, as demonstrated in figure (16).

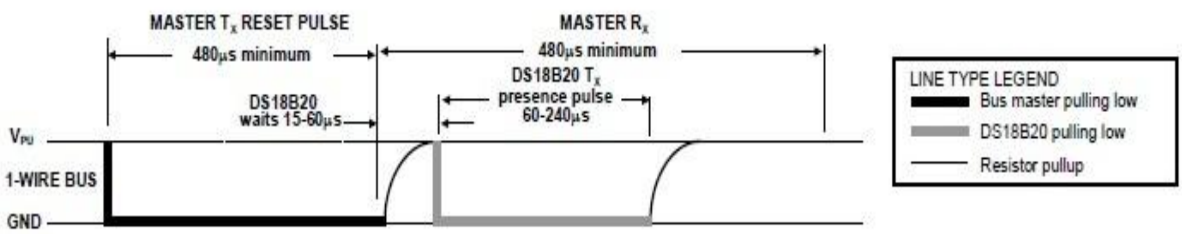


Figure (16): Initialization timing [9]

3.4 SELECTION AND DESCRIPTION OF ETHERNET MODULE

The ENC28J60 has been designed to operate as an Ethernet network interface for any controller fitted-up with SPI (*Serial Peripheral Interface*). It is a stand-alone Ethernet controller with a popular link SPI. It meets all of the IEEE 802.3 standards, and includes a number of packet filtering scheme to limit incoming packets. It also supplies an internal DMA module for hardware assisted checksum calculation and fast data throughout, which is used in various network protocols.

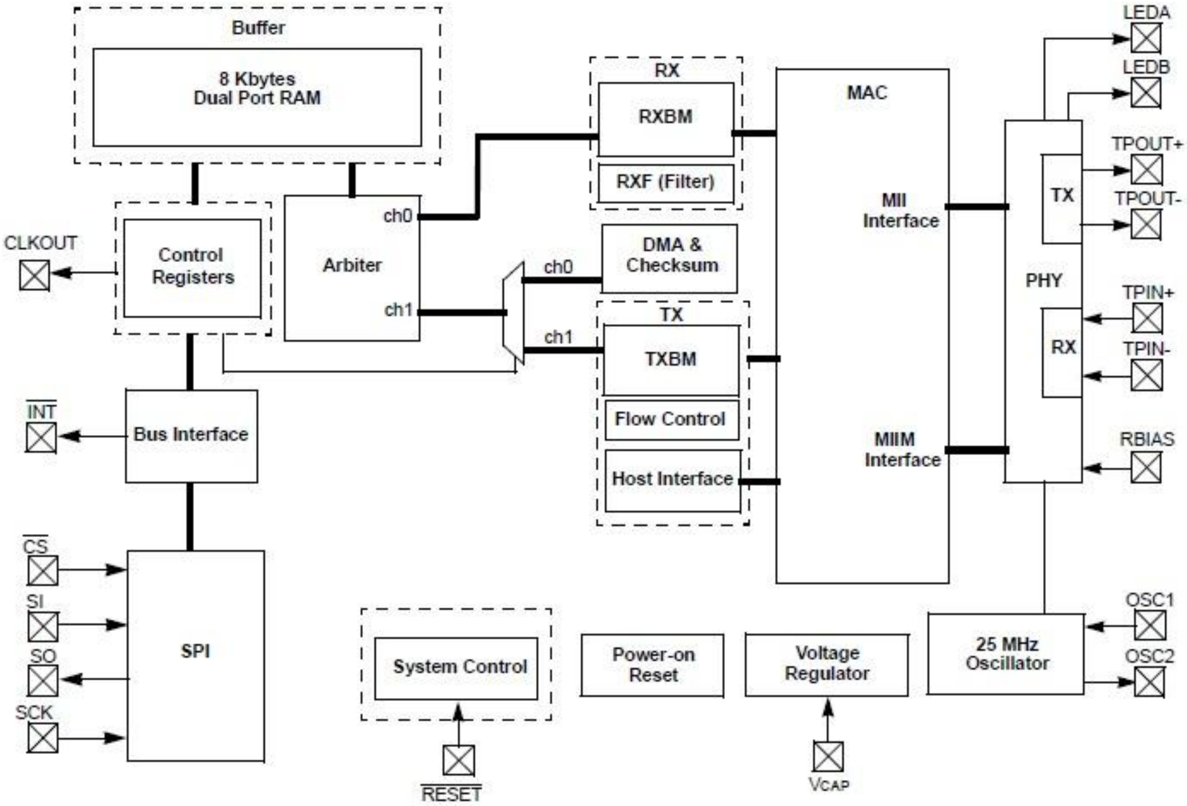


Figure (17): Block diagram of ENC28J60 [10]

A communicate with host controller is coming true via an interrupt pin and the SPI, with clock rates of up to 20MHz. Two dedicated pins are used for LED link and network activity indication. In figure (17) is shown a simple block diagram of the ENC28J60. All that are required to connect a microcontroller to an Ethernet network are two pulse transformers and a few passive components as demonstrated in figure (18).

Seven major functional blocks composing the ENC28J60 are mentioned below (Other support blocks are contained in the device, such as the oscillator, on-chip voltage regulator, level translator to provide 5V tolerant I/Os and system control logic):

1 Serial peripheral interface that serves as a communication channel between the host controller and the ENC28J60.
2 Control registers which are used to control and monitor the ENC28J60.
3 Dual port RAM buffer for received and transmitted packets.
4 An arbiter to control the access to the RAM buffer when requests are made from DMA, transmit and receive blocks.
5 The bus interface that interprets data and commands received via the SPI interface.

6 The MAC module that implements IEEE 802.3 compliant MAC logic.
7 The PHY module that encodes and decodes the analog data that is present on the twisted pair interface.

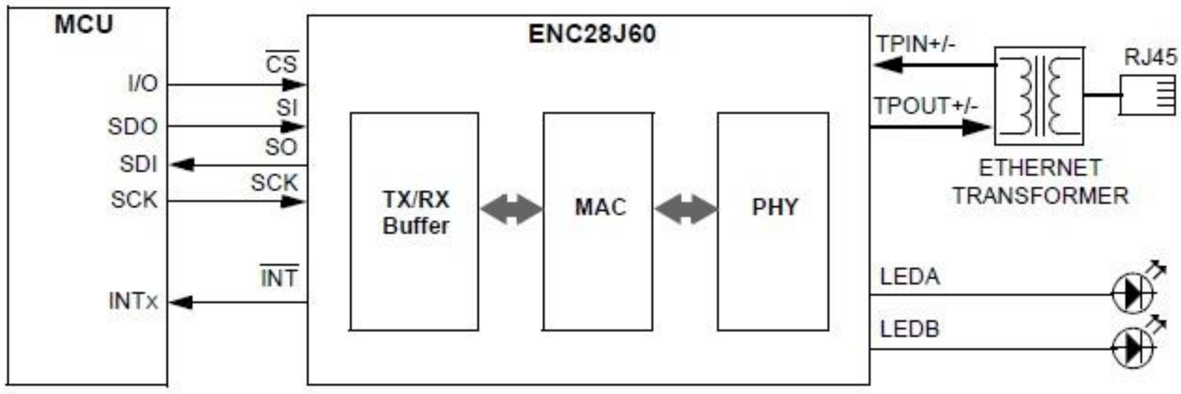


Figure (18): The connection of ENC to microcontroller [10]

AND ETH v2 module based on Microchip ENC28J60 distribution with a popular link SPI. SPI is used for bidirectional synchronous serial data interface between a system and the microcontroller. SPI link consists of two lines that transmit data synchronously in both directions (SI and SO) and the clock signal SCK line.

ENC28J60 system is assigned its own MAC (*Medium Access Control*) address, which leads it to the Physical Layer PHY. The AND ETH v2 module uses an integrated RJ45 network, including the isolation transformer and two LEDs that show the connection and data transfer.

Basic features of AND ETH v2:

- Based on ENC28J60
- Power supply : 3.3V or 5V
- The input buffer capable of operating with 5V powered systems
- RJ45 contains isolation transformer and two LEDs to indicate network connection status
- Networking at a speed of 10Mb/s

3.5 TECHNOLOGY OF POWER OVER ETHERNET (POE)

Power over Ethernet (PoE) or Active Ethernet eliminates the need to run VAC power to Wireless Access Points and other devices on a wired LAN. Using Power over Ethernet system installers need to run only a single CAT5 (*Category 5 cable*, for high power levels, but for low power levels it can operate with *Category 3 cable*) Ethernet cable that carries both electrical power safely, along with data to each device.

Power is supplied in common mode over two or more of the differential pairs of wires found in the Ethernet cables and comes from a power supply within a PoE enable networking device such as an Ethernet switch or can be injected into a cable run with a mid-span power supply.

There are two standards for the use of PoE, comparison between these two standards is shown in the table below:

- The original one is IEEE 802.3af-2003 PoE standard, which provides up to 15.4W of DC power to each device, this means, it works with minimum of 44VDC and 350mA. Only 12.95W is assured to be available at the powered device as some power is dissipated in the cable.
- The updated IEEE 802.3at-2009 PoE standard that is also known as PoE plus (PoE+), provides up to 25.5W of power. Powering a device by using all four pairs for power is prohibited in this standard.

Table (3): Standard PoE parameters and comparison

Property	**802.3af**	**802.3at**
Power available at PD	12.95 W	25.50 W
Max power delivered by PSE	15.40 W	34.20 W
Voltage range (at PSE)	44 – 57 V	50 – 57 V
Voltage range (at PD)	37 – 57 V	42.50 – 57 V
Maximum current	350 mA	600 mA
Maximum cable resistance	20 Ω	12.5 Ω
Power management	Three power class levels negotiated at initial connection	Four power class levels negotiated at initial connection or o.1W steps negotiated continuously
Support cabling	Category 3 and Category 5	Category 5
Support modes	Mode A (end-span), Mode B (mid-span)	Mode A (end-span), Mode B (mid-span)

Figure (19): PoE splitter (5V, 9V, 12V)

3.6 COMPARISON POE WITH USB

PoE provides both data and power connections in one cable, so equipment doesn’t need a separate cable for each need. For equipment that does not already have a power or data connection, PoE can be attractive when the power demand is modest. It can provide long cable runs (e.g. 100meters) and can deliver 12W galvanic isolated power. PoE can be used for IP telephones, cameras with zoom, remote Ethernet switches, industrial devices (sensors, controls, meters) and access control and help points (keyless entry, entry card).

There are many competing technologies, here are described some of them. The USB (*Universal Serial Bus*) provides both data and power, but it can use only short cables, for maximum length e.g. 5 meters, and provides less than 2.5W of non-isolated power. It is cheaper than PoE and works well for low power peripherals such as a computer mouse, headphones, microphone or a serial port.

PoE can have advantages over other technologies, depending on the application:

- Peer to peer network access, when a device is connected to the network, it is accessible to many users
- Modest power
- Fast data rate
- Inexpensive cabling

POWER SOURCING EQUIPMENT: Power Sourcing Equipment (PSE) in a device such as an Ethernet switch or hub, that supplies power over the twisted pairs and controls the detection, classification, and shutdown activities of a powered device at the other end of the cable. The maximum allowed continuous output power per cable in IEEE 802.3af is 15.4W, and 25.5W for the later specification, IEEE 802.3at.

POWERED DEVICE: Power Device (PD) is a device, such as a wireless access point, that draws power from a PSE. As shown in the table below, there are four classes of PDs defined by 802.3af, each with a different power rating (12.95 W being the maximum). IEEE 802.3at increases this limit up to 59 W if all four twisted pairs are in use.

MID-SPAN: Mid-span is a type of PSE that acts as an intermediary between a non PoE switch and PD by injecting power into the cable. Mid-span provides an easy means of incorporating PoE into an existing network, because there is no need to replace any existing switches or infrastructure.

Table (4): Power levels available

Class	Usage	Classification current [mA]	Power range [W]	Class description
0	Default	0 - 4	0.44 – 12.94	Classification unimplemented
1	Optional	9 - 12	0.44 – 3.84	Very low power
2	Optional	17 - 20	3.84 – 6.49	Low power
3	Optional	26 - 30	6.49 – 12.95	Mid power
4	Valid for 802.3at devices, not allowed for 802.3af devices	36 - 44	12.95 – 25.5	High power

POWERING DEVICES: For powering of devices there are two modes available, A and B. on the data pairs of 100BASE-TX or 10BASE-T, mode A delivers power. Mode B delivers power on the spare pairs. In case there are no spare pairs and all power is delivered using the phantom technique, there can be used 1000BASE-T Ethernet for PoE.

In mode A there are two alternate configurations, MDI and MDI-X, using the same pairs but with different polarities. In mode A, pins 1 and 2, which is the 2.pair of T568B wiring, from one side 48 VDC, and from the other side are pins 3 and 6, that are the 3.pair of T568B. These two pairs are used for data transmission in 10BASE-T and 100BASE-TX, which allows the provision of both data and power over only two pairs in such network. In mode B, the 1.pair which includes pins 4 and 5 in both T568A and T568B wiring, from one side of the DC supply and 4.pair with pins 7 and 8 provide the return, these are the spare pairs in 10BASE-T and 100BASE-TX. Therefore, mode B requires a 4 pair cable.

The PSE decides if power mode A or B should be used, it can implement mode A or mode B or both. PDs that implement only mode A or mode B are disallowed by the standard. A PD indicates that it is standards compliant by placing a 25kΩ resistor between the powered pairs. In case PSE detects a resistance that is too high, power is not applied. This made as a protection for devices that do not support PoE. An optional power class feature allows the PD to indicate its power requirement by changing the sense resistance at higher voltage. To stay powered, the PD must continuously use 5-10mA for at least 60 ms with no more than 400ms since last use or else it will be unpowered by the PSE.

PSE has two types, end-spans and mid-spans. End-spans or as called PoE switches, are Ethernet switches which includes the power over Ethernet transmission circuitry. Mid-spans are power injectors that stand between a regular Ethernet switch and the powered device, injecting power without affecting the data. End-spans are normally used on new installations or when the switch has to be replaced for other reasons, which makes it convenient to add the PoE capability. Mid-spans are used when there is no desire to replace and configure a new Ethernet switch, and only PoE needs to be added to the network.

In the table below, are shown the differences between various wiring of modes A and B in the 802.3af standard:

Table (5): 802.3af standards A and B [10]

Pins on switch	**10/100DC on spares (mode B)**	**10/100DC & data (mode A)**	**1000 DC & Bi data (mode B)**	**1000 DC & Bi data (mode A)**
Pin 1	Rx +	Rx + ;DC +	TxRx A +	TxRx A + ; DC+
Pin 2	Rx -	Rx - ;DC +	TxRx A -	TxRx A - ; DC+
Pin 3	Tx +	Tx + ;DC -	TxRx B +	TxRx B + ; DC-
Pin 4	DC +	Unused	TxRx C + ; DC+	TxRx C +
Pin 5	DC +	Unused	TxRx C - ; DC+	TxRx C -
Pin 6	Tx -	Tx - ;DC -	TxRx B -	TxRx B - ; DC-
Pin 7	DC -	Unused	TxRx D + ; DC-	TxRx D +
Pin 8	DC -	unused	TxRx D + ; DC -	TxRx D -

4 SOFTWARE DESCRIPTIONS

4.1 FLOWCHART OF THE SYSTEM

The whole system and its function is simply shown if a flowchart in figure (20). Where in the beginning is necessary to configure the chip for temperature sensing, which will be unleashed into an endless *WHILE* loop. If the main program cycle has not been run, the chip does not respond and thus is the end.

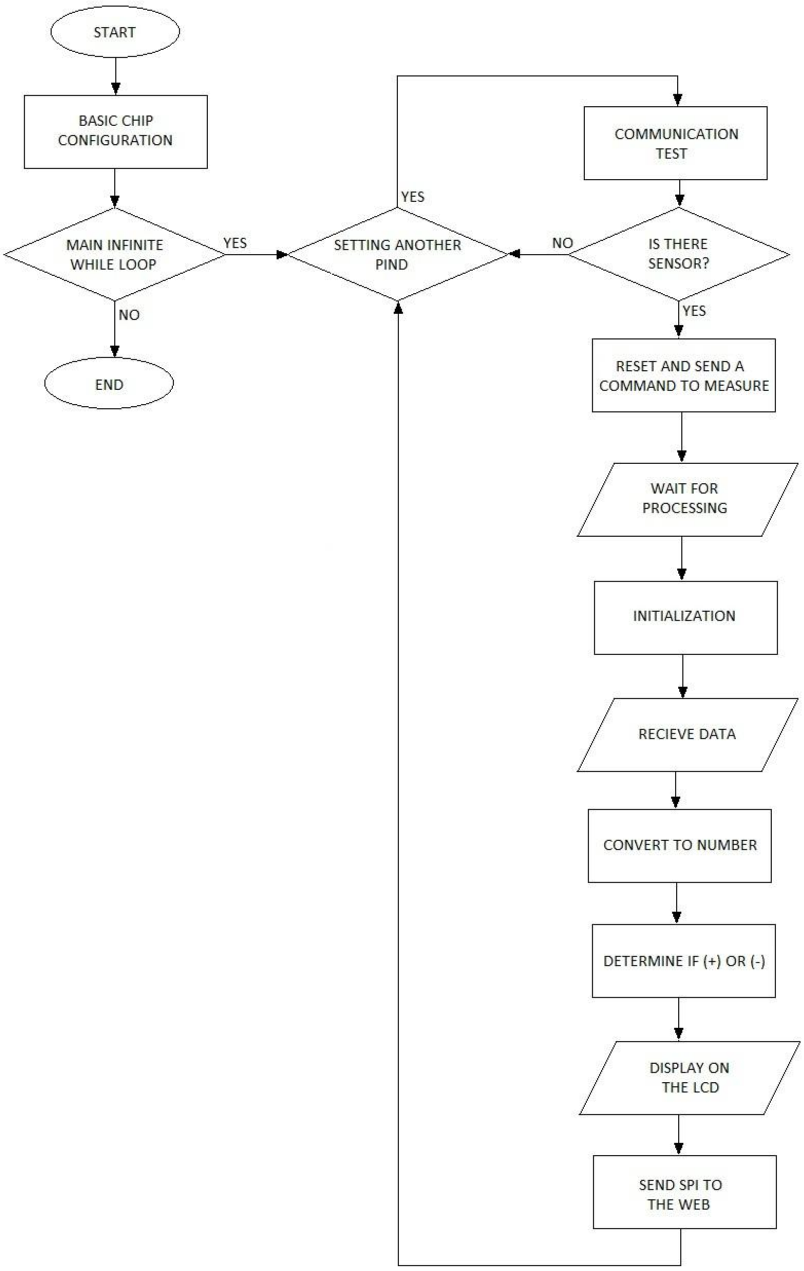


Figure (20): Floatchart of the system

When the program starts, a communication test takes place on PORTD to find on which PIN is connected a sensor. This test continues until recognizing of connection. If sensor is connected and responding, perform of resetting is executed for correct start of communication and temperature sensing, then a command of measuring is sent from the microcontroller to the sensor. After processing, the sensor should be initialized to be able to receive data of measured values.

The acquired values are in a binary form; therefore, they have to be converted into decimal form, to let them appear on LCD and be sent to the WEB. It is also important to determine the sign of measured data to distinguish positive from negative values. After finishing the whole process, comes the final step of values depiction on the LCD and sending them through SPI to the WEB.

Sensor initialization and delays are explained in figure (21)

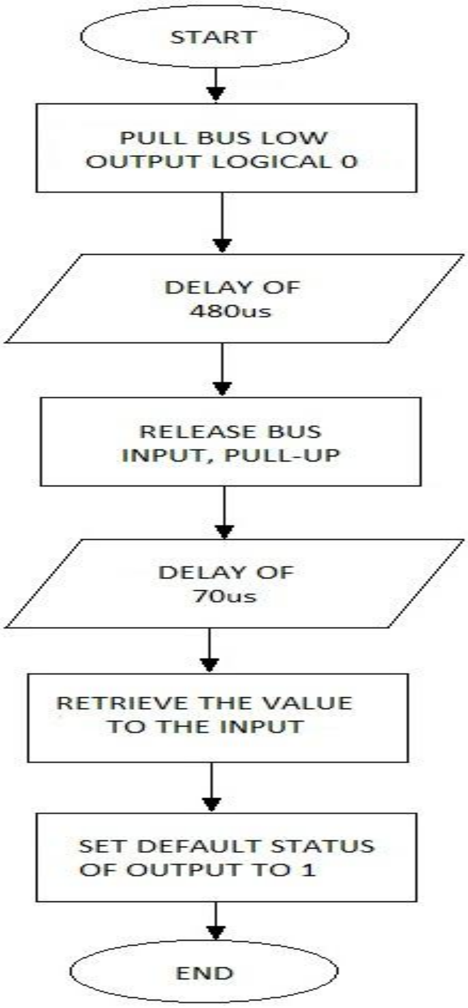


Figure (21): Floatchart of sensor initialization

4.2 COMPONENT PARTS OF CREATED PROGRAM

In this section of work are described the component parts of the created program, which shows how to initialize the LCD (*dot matrix Liquid Crystal Display*), initialization of the temperature sensors, the storage of measured values and their transmission. Then is demonstrated the communication with Ethernet module ENC28J60. And in the last part is described HTML code, where the values are demonstrated after sending them via Ethernet.

4.2.1 DISPLAY INITIALIZATION

The use of LCD was selected to facilitate the control on program and its proper operation. As well as, LEDs have been added. For the proposed system was chosen a display character, which is commercially available includes HITACHI HD44780 LCD controller of 2 x 16 (2 lines of 16 columns, figure (22)). Their advantage is that they operate the same way, even if they are from different manufacturers. For communication with LCD controller can be used 8 - bit or 4 - bit communication for data and 3 control pins.

8-bit communication is faster, but takes up to 11- pin microcontroller, while the 4-bit communication is dividing the data word into two Niblos and sends them in time-multiplex. Imaging is slower, but it reduces the number of conductors. Data transfer between the microcontroller and the display controller is controlled by three signals [11]:

- RS = 0 instruction transmission, RS = 1 data transmission
- RW = 0 write data/instructions to the LCD, RW = 1 read data/instructions from LCD
- E - high level pulse triggers communication

To start communication with the display, it has been foremost initialized through initialization lines and file shown below:

- alcd.h – library for communication with LCD
- lcd_init(16) – initialize the LCD for 2 lines and 16 columns
- lcd_gotoxy(0,0) – goes on the second LCD line
- lcd_putsf'("Teplota:") – display the message in parentheses

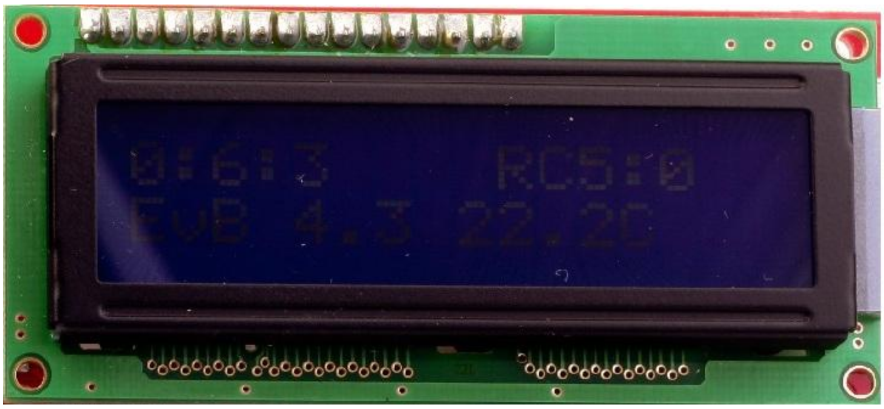


Figure (22): HITACHI HD44780 LCD controller (2x16)

4.2.2 SENSOR INITIALIZATION

An initialization sequence is needed for all operations on the 1-wire bus. The initialization starts with a reset pulse transmitted by the bus master followed by presence pulse sent by the slave. This presence pulse lets the bus master know that slave device, in this case sensor DS18B20, is on the bus and ready to start operating. The master device must issue an appropriate ROM command before issuing a DS18B20 function command. In this work is used one ROM command:

- **SKIP_ROM [CCh]** – This command is used by the master to address all devices on the bus simultaneously without sending out any ROM code information. It has to be filled with (CCh) protocol.
- While using this command there should be following Function commands as described below:
- **CONVERT_T [44h]** – This temperature conversion command initiates temperature conversion, where DS18B20 transmits conversion status to master, it contains (44h) protocol.
- **READ_SCRATCHPAD [BEh]** – This reads the entire scratchpad including the CRC byte, where DS18B20 transmits up to 9 data bytes to master, it contains (BEh) protocol.
- **WRITE_SCRATCHPAD [4Eh]** – This command wires data into scratchpad bytes 2, 3, and 4 (T_H, T_L, and configuration registers), it contains (4Eh) protocol.
- **EE_WRITE [48h]** – This command copies T_H, T_L, and configuration register data from the scratchpad to EEPROM, it contains (48h) protocol.
- **READ_POWER_SUPPLY [B4h]** – This signals DS18B20 power supply mode to the master, by using (B4h) protocol.

Illustration from the source code:

```
#define SKIP_ROM 0xCC
#define CONVERT_T 0x44
#define READ 0xBE
#define WRITE 0x4E
#define EE_WRITE 0x48
#define READ_PS 0xB4
```

The table below shows the relationship between temperature and data:

Table (6): Convertion digital outout (Binary and HEX) into decimal system [9]

TEMPERATURE (°C)	**DIGITAL OUTPUT (BINARY)**	**DIGITAL OUTPUT (HEX)**
+125	0000 0111 1101 0000	07D0h
+85	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

Illustration from the source code for initialization of the DS18B20, reading a bit from 1-wire and writing a bit on it:

```
unsigned short w1_init0(unsigned short pin) // initialization of 1-wire
    {
      Detect = 1;
      DDRD  |= (1<<pin);        // pull bus low  (output logical 0)
      PORTD &= ~(1<<pin);       // PORTD changes from high to low
      delay_us(480);            // delay of 480us
      PORTD |= (1<<pin);        // PORTD changes from low to high
      DDRD  &= ~(1<<pin);       // Release bus (input, pull-up)
      delay_us(70);             // delay of 70us

      detect = !(PIND & (1<<pin));  // if 0,sensor is detected,
                                     if 1, sensor is not detected

      PORTD |= (1<<pin);        // PORTD changes to high
      DDRD  |= (1<<pin);        // pull bus low  (output logical 0)
     delay_us(410);
     return detect;
    }
```

```
unsigned short w1_read0(unsigned short pin) // reading 1bit of 1wire
{
   Result = 0;

   DDRD |= (1<<pin);       // pull bus low (output logical 0)
   PORTD &= ~(1<<pin);     // PORTD from high to low
   DDRD &= ~(1<<pin);      // Release bus (input pull-up)

   delay_us(10);           // delay of 10us for "ananswer" from 1-wire
   result = PIND.6;        // left-shift once
   delay_us(30);           // delay of 30us

   PORTD |= (1<<pin);      // PORTD from low to high
   DDRD  |= (1<<pin);      // pull bus low (output logical 0)
   return result;

 }

void w1_write0( unsigned short pin, unsigned short thebit) //write 1bit
                                                              on 1wire
{
   if(thebit == 1) {
     PORTD  &= ~(1<<pin);       // from high to low
     //PORTD=0x00;
     delay_us(6);
     PORTD  |= (1<<pin);        // release the bus (input pull-up)
     //PORTD=0xFF;
     delay_us(60);
   }
   else {
     PORTD  &= ~(1<<pin);

     delay_us(60);              // delay "low" for 60us
     PORTD  |= (1<<pin);        // release the bus (input pull-up)
     //PORTD=0xFF;
     delay_us(6);
   }
 }
```

4.2.3 STORAGE OF MEASURED VALUES

The storage of measured values serve's for two matters. The first one is to send and image them on the LCD, in order to control and be sure of proper function measurement. The second and the main matter, is to move the values into to HTML (*Hyper-Text Markup Language*) code in order to send via Ethernet and demonstrate results on a PC in a web page.

After initialization the DS18B20, and before the storage of data, is necessary to retrieve the whole byte from the 1-wire. This step is made by repetition of reading and writing from the previous section for 8 times as below:

```
void w1_writebyte0(unsigned short pin, unsigned short command)  //write
                                                        1byte from 1-wire
 {
   unsigned char temp,i;

   for(i=0;i<8;i++) {
     temp = (command & 0x01);   // temp is equal to the value of the
```

```
                                    first bit of command LSB
     if(temp)                   //if true
        w1_write0(pin,1);       //write bit1
     else
       w1_write0(pin,0);        //write bit0
     command = command >> 1;    //right shift by 1 so next bit comes in
LSB
   }
 }

unsigned short w1_readbyte0(unsigned short pin)        //read 1byte
{

   int i;
   temp = 0x00;
   for(i=0;i<8;i++)
     temp |= ((w1_read0(pin)) << i); //read bit by bit and keep
             shifting left to reach MSB so when i=7 MSB will have bit 7
   return temp;
 }
```

4.2.4 COMMUNICATION WITH ETHERNET MODULE

For starting communication with Ethernet module ENC28J60, must be added to the main program some header files, showed below, that allows commencement of this communication:

```
#include "ip_arp_udp_tcp.h"
#include "enc28j60.h"
#include "timeout.h"
#include "avr_compat.h"
#include "net.h"
```

This header files inform the microcontroller of communication with Ethernet module type ENC28J60, that there will be some interruptions and timeout, sending some information to Internet, and transmission protocols TCP, UDP, for predetermined value of IP.

The communication between the microcontroller and the Ethernet module uses a communication file *enc28j60.C*, which is created by **Pascal Stang** [12], and recorded to the added CD.

4.2.5 HTML CODE FOR WEB SERVER

The measured temperature values, should be moved or shifted from the variable were saved in, to the variable in the HTML code, included in the main program, before imaging. This movement of values comes after specifying serial number (from 1 to 8) of temperature sensor, where according to this number, the value line up into the correct field in the HTML code.

After filling fields with correct values, the HTML code is sent through Ethernet module, and result imaging take place for predetermined IP and MAC address, as demonstrated in figure (24).

5 RESULT EVALUATION

Software for microcontroller is written in the programming language C. it was created and debugged in a software development **Code-Vision-AVR V2.05.0 Professional**, (it has an integrated development environment, with an automatic program generator and In-System programmer for the Atmel AVR family of microcontrollers), and tested on the development kit AND-Tech EvB 4.3 rev.4.

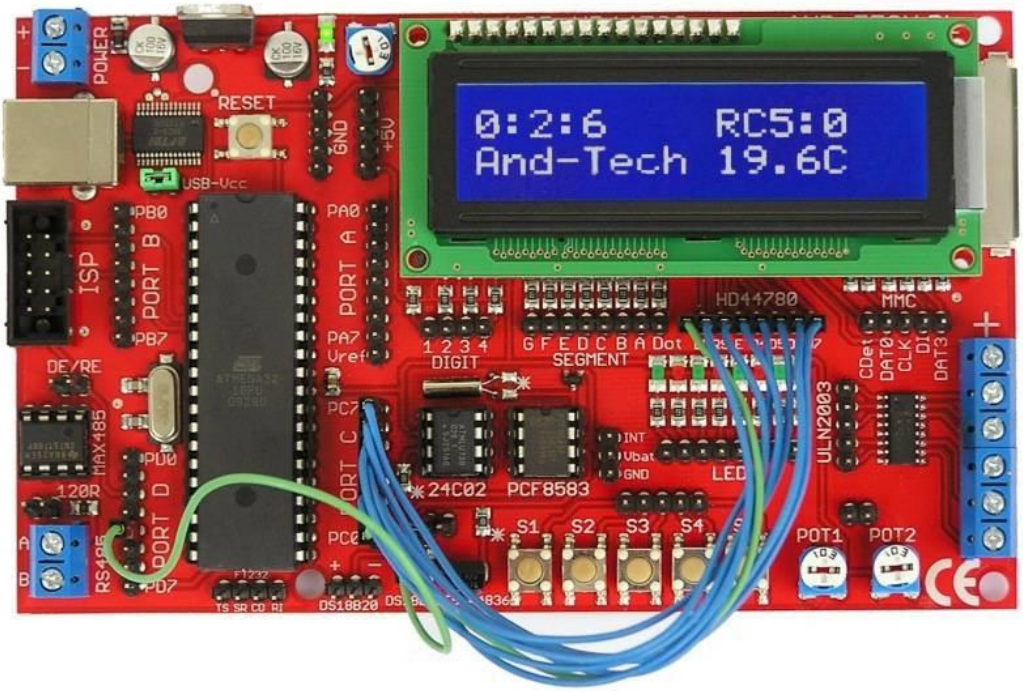


Figure (23): kit AND-Tech EvB 4.3 rev4 [13]

The whole program is designed as one large part with no dividing into smaller parts. It is gradually written by the function description in section 3.1.

This program does not use interrupt; this was made for progressive measure and data imaging and sending, which makes the whole chain of events connected to each other. The complete program for the project is available on CD attached to work, in a form of AVR studio 4.

Before describing the program will be presented peripherals and features of a microcontroller. The chosen one is from Atmel, model ATmega644, with encapsulation of DIP40. In the development board was equipped with a 16MHz crystal, which was tested for timing delays and left on the final product.

Electrical wiring diagram, added to the appendixes part, shows that the microprocessor contains 4 input and output ports (*A, B, C, D*). In the program are used pins for 8 temperature sensors DS18B20 connected on port D, port C is connected to the HITACHI HD44780 LCD, and port B is used for connection with the Ethernet module ENC28J60, which has bus interface that interprets data and commands received via the SPI interface. Using this Ethernet module makes possible of powering the whole system by the use of PoE technology, supplied by the 5V that is sufficient to run all the peripherals.

Flowchart of the program is shown in part (4.1) figure (20), where are describer the requirements for start-up and functionality of the program. Flowchart for initialization of the sensor is shown and explained in part (4.1) figure (21).

Values displayed on the LCD prove the correct functionality of the program. Requirement for temperature measurement accuracy of at least $0.5^{o}C$ is applied, and in the program for implementation has been deployed to $0.1^{o}C$, which is seen on the LCD as one position after the decimal point.

Storage and sending data via Ethernet serves to monitor the temperature over a website on internet, with no need to watch the hardware and LCD. This gives the user an opportunity of monitoring temperature of some different places at the same time from any place, and from any PC connected to Internet, in case of having the predefined IP and MAC address.

HTML code for used in this work is added to the appendixes part, and demonstrated in figure below, where for each position by the number should be sent measured value from a sensor with the same position number on the port. This way, the user can recognize which sensor is connected, and which is not, and the displayed value is sent from which sensor.

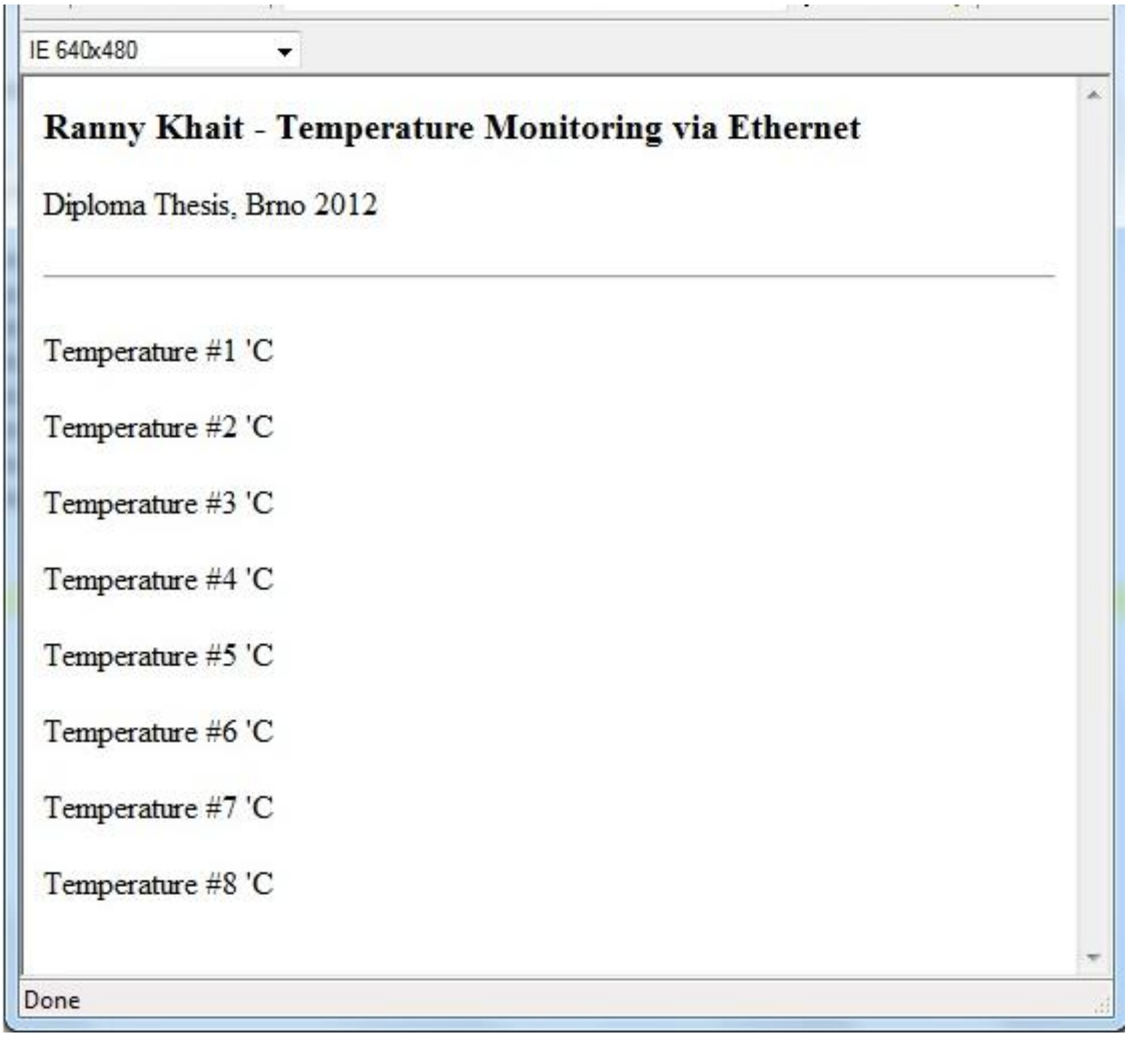


Figure (24): Temperature monitoring on web

CONCLUSION

In this thesis, a research is performed on temperature monitoring and sensing, and there are discussed a variety of sensors including integrated circuits for temperature measurement. For the purpose of this work is chosen an integrated sensor that has serial digital output of the measured values. For the thesis was assigned a requirement on the accuracy of 0.5°C, were selected sensors, which allow better accuracy, the device makes possible to display the desired temperature range with an accuracy of 0.1°C.

Another part is devoted to technology on IEEE 802.3 standard called Ethernet, frames and packets description, protocol TCP/IP and collision resolving. In the practical part is designed a system for temperature measurement and its transmission through the Ethernet network, then discourse is made discussion and choice of a suitable microcontroller and Ethernet module for communication between microcontroller and Ethernet network. System powering is solved using PoE technology, through data cable itself. An electrical diagram and component part-list are created based on the block diagram.

The program for a microcontroller is constructed in C language; it enables screening messages on the kit's display. The program communicates with the connected sensors, and during the running it can update and provide treatment of integration of new sensors or disconnection of the existing ones. The program allows handling up to 8 sensors simultaneously. It also includes a simple Web page to interpret the temperature for the user. Using this program, microcontroller communicates with the Ethernet module, by defining the MAC address and IP address of the interface.

The correctness of the design was tested using development kit for microcontroller ATmega644p, and connected with Ethernet module ENC28J60. For screening temperature values, is used alphanumeric display with Hitachi's controller. After connecting the sensor to the input of microcontroller, correct temperature value is displayed. The temperature of each individual sensor is displayed on the display for 2 seconds, then is displayed the temperature of the following sensor. A program for Ethernet module is created, the emulator development environment simulates correct communication, but the output of the Ethernet module is not communicating. While testing the communication by using a program in BASCOM language, the module communicates. The problem will probably be in a header file for C language. Therefore, values transmission could not be fully realized, but it is functional to the output of the microcontroller including. Beyond this point objectives were fully met.

REFERENCES

[1] RIPKA, P, et al. Senzory a převodníky. Dotisk prvního vydání. Praha : ČVUT v Praze, 2005. 135 s. ISBN 80-01-03123-3.

[2] KREIDL, M. Měření teploty - Senzory a měřicí obvody. 1. vyd. Praha : BEN - technická literatura, 2005. 239 s. ISBN 80-7300-145-4.

[3] U.S. Sensor [online]. C2010 [2011-11-20]. NTC Thermistor. Available from: <http://www.ussensor.com/prod_Probes_NTC.html>.

[4] HW.cz [online].2004 [2012-01-05]. Měření teploty – polovodičové odporové senzory teploty. Available from: <http://hw.cz/Teorie-apraxe/ Dokumentace/ART1141-Mereni-teploty—polovodicove-odporovesenzory- teploty.html>.

[5] DOSTÁLEK, L. Velký průvodce protokoly TCP/IP a systémem DNS. 1 TH edit. Praha 2000. 435 p. ISBN: 80-7226-323-4.

[6] DOYLE, P. Intoduction Real – Time Ethernet 1. [PDF]. [2009-04-05]. Available from: <http://www.datalinkcom.net/Real%20Time%20Ethernet1.pdf>.

[7] DEVASAHAYAM, S.R. Signals and Systems in Biomedical Engineering. Signal Processing and Physiological Systems Modeling. Kluwer Academic 2000

[8] ATmega644P datasheet [online]. Atmel, 2010 [2011-12-09]. Available from: <http://www.atmel.com/dyn/resources/proddocuments/doc8011.pdf>.

[9] DS18B20 datasheet [online]. Maxim, 2008 [2011-11-15]. Available from: <http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf>.

[10] IEEE 802.3AF Power Over Ethernet, [2012-01-22]. Available from: <http://www.siemon.com/us/standards/13-9_ieee_802_3af_power_over_ethernet.asp>

[11] Display HD44780 datasheet [online]. Hitachi [2012-02-13]. Available from: <http://www.alldatasheet.com/datasheet-pdf/pdf/63673/HITACHI/HD44780.html>.

[12] AVRlib [online]. Pascal Stang [2012-04-18]. Available from: <http://www.procyonengineering.com/>

[13] Zestaw startowy EvB 4.3 [online]. 2011 [2011-11-24].Firma And-Tech . Availabe from: <http://www.and-tech.pl/nasza-oferta/95-zestaw-startowyevb-43>.

[14] Datasheet, Operating Instructions for kit EvB 4.3 v4 [online]. 27-08-2010 [2011-11-24]. And-Tech Zestaw startowy EvB 4.3. Available from: <http://www.and-tech.pl/EvB4.3/Instrukcja_EvB_4.3v4.pdf>.

[15] MATOUŠEK, David. Práce s mikrokontroléry Atmel AVR: ATmega16. Praha: BEN, 2006. 319 s. ISBN 80-7300-178-8.

[16] Telemeter Electronic s.r.o. [online]. 2010 [2012-04-02]. Datasheet thermistor sensors. Available from: <http://www.telemeter.info/documents/ntc_termistory.pdf>.

[17] Omega Engineering [online]. 2010 [2012-04-02]. Precizní termistorová čidla. Availale from: <http://www.omegaeng.cz/ppt/pptsc.asp?ref=44000_THERMIS_ELEMENTS &Nav=temd01>.

[18] Power over Ethernet [online]. [2012-02-14]. Available from: <http://www.lindy.co.uk/power-over-ethernet-splitter-5v-9v-12v-dc/25055.html>

[19] Vanek,T. Ethernet, Síťové technologie 1, lecture 3. [pdf]. [2011-12-09]Available from: <https://dsn.felk.cvut.cz/wiki/vyuka/cviceni/x326st1/x326st1>

ABBREVIATIONS

C - Celsius

K - Kelvin

QoS - Quality of Service

UDP - User Data Protocol

TCP - Transmission Control Protocol

IP - Internet Protocol

MAC - Medium Access Control

RT - Real Time

LED - Light Emitting Diode

LCD - dot matrix Liquid Crystal Display

AD - Analog to Digital

PC - Personal Computer

PoE - Power over Ethernet

SPI - Serial Peripheral Interface

IEEE - Institute of Electrical and Electronics Engineers

LAN - Local Area Network

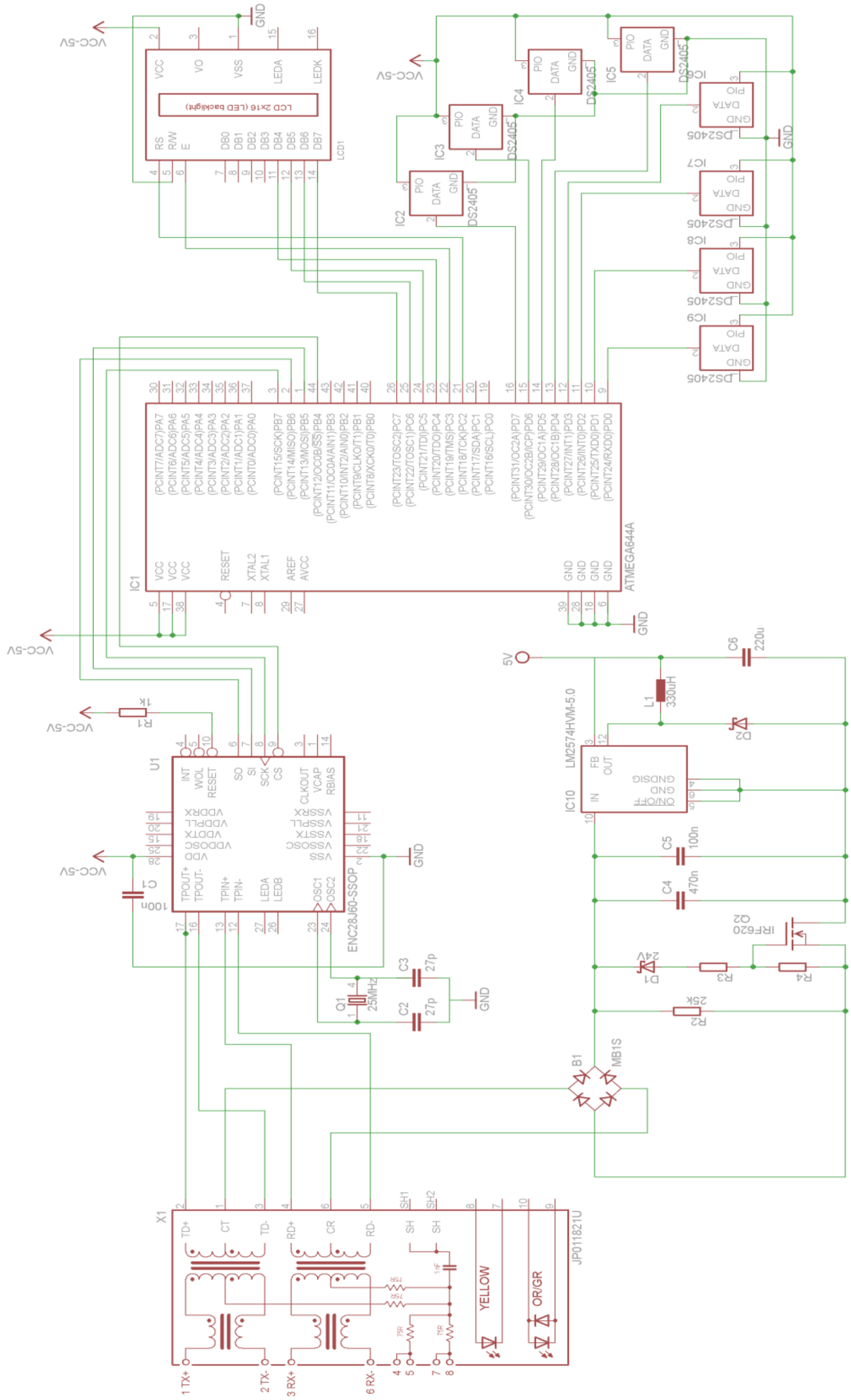
HTML - HyperText Markup Language

APPENDIXES

HTML CODE

```
<html>

<head>

<title>Ranny Khait</title>

</head>

<h3>Ranny Khait - Temperature Monitoring via Ethernet</h3>

<p>Diploma Thesis, Brno 2012</p>

<hr>

<p> Temperature #1 <temp-1> 'C </p>

<p> Temperature #2 <temp-2> 'C </p>

<p> Temperature #3 <temp-3> 'C </p>

<p> Temperature #4 <temp-4> 'C </p>

<p> Temperature #5 <temp-5> 'C </p>

<p> Temperature #6 <temp-6> 'C </p>

<p> Temperature #7 <temp-7> 'C </p>

<p> Temperature #8 <temp-8> 'C </p>

</body>

</html>
```

ELECTRIC WIRING DIAGRAM



PART-LIST

PART	VALUE	DEVICE	PACKAGE	LIBRARY
B1	MB1S	MB1S	SOIC-4	Rectifier
C1	100n	C2.5/2	C2.5-2	Capacitor-wima
C2	27n	C2.5/2	C2.5-2	Capacitor-wima
C3	27n	C2.5/2	C2.5-2	Capacitor-wima
C4	470n	C2.5/2	C2.5-2	Capacitor-wima
C5	100n	C2.5/2	C2.5-2	Capacitor-wima
C6	220n	C2.5/2	C2.5-2	Capacitor-wima
D1	24V	MBRS320T3	SMC	Diode
D2		MBRS320T3	SMC	Diode
IC1	ATMEGA644	ATMEGA644	TQFP44	Atmel
IC2	DS1820	DS1820	TO92	Maxim
IC3	DS1820	DS1820	TO92	Maxim
IC4	DS1820	DS1820	TO92	Maxim
IC5	DS1820	DS1820	TO92	Maxim
IC6	DS1820	DS1820	TO92	Maxim
IC7	DS1820	DS1820	TO92	Maxim
IC8	DS1820	DS1820	TO92	Maxim
IC9	DS1820	DS1820	TO92	Maxim
IC10	LM2574HVM-5	LM2574HVM-5	SO14W	Linear
L1	330uH	BML15HB121SN1	0402	Inductors
LCD1	LCD_2X16	LCD_2X16	HY-1602F	LCD_2X16_led_backlight
Q1	25MHz	86SMX	86SMX	Crystal
Q2	IRF620	IRF620	TO220	Transistor-fet
R1	1k	R-EU_0207/5V	0207/5V	Rezistor
R2	25k	R-EU_0207/5V	0207/5V	Rezistor
R3	4.7k	R-EU_0207/5V	0207/5V	Rezistor
R4	4.7k	R-EU_0207/5V	0207/5V	Resistor
U1	ENC28J60	ENC28J60	SSOP28	ENC28J60
X1	JP11821U	JP11821U	JP	con-pulse